Spectral Geometry for Structural Pattern Recognition

HEWAYDA EL GHAWALBY

Ph.D. Thesis
This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

THE UNIVERSITY of York
Department of Computer Science
United Kingdom

May 2011

Abstract

Graphs are used pervasively in computer science as representations of data with a network or relational structure, where the graph structure provides a flexible representation such that there is no fixed dimensionality for objects. However, the analysis of data in this form has proved an elusive problem; for instance, it suffers from the robustness to structural noise. One way to circumvent this problem is to embed the nodes of a graph in a vector space and to study the properties of the point distribution that results from the embedding. This is a problem that arises in a number of areas including manifold learning theory and graph-drawing.

In this thesis, our first contribution is to investigate the heat kernel embedding as a route to computing geometric characterisations of graphs. The reason for turning to the heat kernel is that it encapsulates information concerning the distribution of path lengths and hence node affinities on the graph. The heat kernel of the graph is found by exponentiating the Laplacian eigensystem over time. The matrix of embedding co-ordinates for the nodes of the graph is obtained by performing a Young-Householder decomposition on the heat kernel. Once the embedding of its nodes is to hand we proceed to characterise a graph in a geometric manner. With the embeddings to hand, we establish a graph characterization based on differential geometry by computing sets of curvatures associated

with the graph nodes, edges and triangular faces.

The second contribution comes from the need to solve the problem that arise in the processing of a noisy data over a graph. The Principal difficulty of this task, is how to preserve the geometrical structures existing in the initial data. Bringing together several, distinct concepts that have received some independent recent attention in machine learning; we propose a framework to regularize real-valued or vector-valued functions on weighted graphs of arbitrary topology. The first of these is deduced from the concepts of the spectral graph theory that have been applied to a wide range of clustering and classification tasks over the last decades taking in consideration the properties of the graph p-Laplacian as a nonlinear extension of the usual graph Laplacian. The second one is the geometric point of view comes from the heat kernel embedding of the graph into a manifold. In these techniques we use the geometry of the manifold by assuming that it has the geometric structure of a Riemannian manifold. The third important conceptual framework comes from the manifold regularization which extends the classical framework of regularization in the sense of reproducing Hilbert Spaces to exploit the geometry of the embedded set of points. The proposed framework, based on the p-Laplacian operators considering minimizing a weighted sum of two energy terms: a regularization one and an additional approximation term which helps to avoid the shrinkage effects obtained during the regularization process. The data are structured by functions depending on data features, the curvature attributes associated with the geometric embedding of the graph.

The third contribution is inspired by the concepts and techniques of the graph calculus of partial differential functions. We propose a new approach for embedding graphs on pseudo-Riemannian manifolds based on the wave kernel which is the solution of the wave equation on the edges of a graph. The eigensystem of the wave-kernel is determined by the eigenvalues and the eigenfunctions of the normalized adjacency matrix and can be used to solve the edge-based wave equation. By factorising the Gram-matrix for the wave-kernel, we determine the embedding co-ordinates for nodes under the wave-kernel.

The techniques proposed through this thesis are investigated as a means of

gauging the similarity of graphs. We experiment on sets of graphs representing the proximity of image features in different views of different objects in three different datasets namely, the York model house, the COIL-20 and the TOY databases.

Contents

List of Figures

List of Tables

acknowledgements

I would like to express my sincere appreciation and gratitude to my supervisor, Prof. Edwin R. Hancock, for the support, encouragements, suggestions, advices and time given to me throughout my PhD. Without his insightful guidance, it would not have been possible for me to complete this thesis. I also thank my assessor, Dr. Adrian G. Bors for his impartial assessment and constructive comments on my work. My sincere thanks also go to the good friends I made in York; especially, I wish to thank the current and the past members of the CVPR group. Without them my experience at the University of York would have not been as enjoyable and pleasant as it has been.

I do offer my especial thanks to my family for their love and support. My husband Ahmed, My son Adelrahman and my daughter Rana; the greatest family I could have ever wished for. I am looking forward to all the great moments we will share together in the future.

Finally, I gratefully acknowledge the financial support of my scholarship from the Egyptian Government.

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references. I hereby give consent for my thesis, if accepted, to be

made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed . . . . . (candidate)

Date . . . . . .

Some of the material contained in this thesis has been previously published by the author. For a complete list of publications, please refer to the next page.

List of Publications

- Hewayda ElGhawalby, Edwin R. Hancock: Graph Embedding Using an Edge-Based Wave Kernel. SSPR/SPR 2010 Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition, LNCS 6218, pp. 60 - 69.

- Hewayda ElGhawalby, Edwin R. Hancock: Graph Regularisation Using Gaussian Curvature. GbRPR 2009 Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition, LNCS 5534, pp. 233 - 242.

- Hewayda ElGhawalby, Edwin R. Hancock: Geometric Characterizations of Graphs Using Heat Kernel Embeddings. IMA Conference on the Mathematics of Surfaces 2009, LNCS 5654, pp. 124 - 142.

- Hewayda ElGhawalby, Edwin R. Hancock: Characterizing Graphs Using Spherical Triangles. IbPRIA 2009 Proceedings of the 4th Iberian Conference on Pattern Recognition and Image Analysis, LNCS 5524, pp. 465 - 472.

- Hewayda ElGhawalby, Edwin R. Hancock: Measuring Graph Similarity Using Spectral Geometry. ICIAR 2008 Proceedings of the 5th interna-

tional conference on Image Analysis and Recognition, LNCS 5112, pp. 517 - 526.

- Hewayda ElGhawalby, Edwin R. Hancock: Graph Characteristic from the Gauss-Bonnet Theorem. SSPR/SPR 2008 Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, LNCS 5342, pp. 207 - 216.

- Hewayda ElGhawalby, Edwin R. Hancock: Graph Clustering using heat kernel embedding and spectral geometry. YDS 2007 Proceedings of the First York Doctoral Symposium on Computing, pp. 19 - 26.

CHAPTER 1

Introduction

1.1 Thesis Motivation

Pattern recognition techniques are concerned with the theory and algorithms of putting abstract objects, e.g., measurements made on physical objects, into categories or classes. In a typical situation the categories are assumed to be known in advance, even though there are techniques to learn the categories (clustering). The aim is to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. Methods of pattern recognition are useful in many applications such as information retrieval, data mining, document image analysis and recognition, computational linguistics, biometrics and bioinformatics. Depending on the application, these objects can be images, signal waveforms or they can be any type of measurements that need to be classified. With a long history back to the eighteenth century till the 1960's, pattern recognition was mostly the output of theoretical research in the area of statistics. The dramatic growth of

using computers increased the demand for practical applications of pattern recognition, which in turn set new demands for further theoretical developments in the underlying algorithms and techniques. Rapidly, pattern recognition becomes an enabling technology in applications as diverse as image and signal processing, remote sensing, data and image compression, surveillance imaging, industrial vision and audio signal processing, medical data processing, and a wide variety of military applications. The development and testing of pattern recognition algorithms becomes a key to mathematics and computer science research.

Many common pattern recognition algorithms are probabilistic in nature, in that they use statistical inferences including generative methods such as those based on Bayes decision theory and related techniques of parameter estimation and density estimation as well as discriminative methods such as nearest-neighbor classification and support vector machines. Similarly, new models based on kernels have had significant impact on both algorithms and applications of pattern recognition. In general, solving pattern recognition problems involves an enormous amount of computational effort. One approach for speeding up the process is to embed the objects from a high-dimensional space (Euclidean or pseudo-Euclidean) into a low-dimensional target space, which is more convenient to operate on and then cluster the objects in that low-dimensional space. In this scenario, we aim in our work here to import methods from both "Spectral Geometry" and "Manifold Learning" to the pattern recognition field.

Spectral Geometry is a field in mathematics which is concerned with characterizing the geometric structures of manifolds by using the spectrum of canonically defined differential operators. In particular, the special case of the Laplace-Beltrami operator on a closed Riemannian manifold has been most intensively studied. Nevertheless, many of the Laplace operators in differential geometry have been examined. In some way, spectral geometry is quite close to spectral graph theory. In this sense, the graph Laplacian matrix is regarded as the discrete approximation to the Laplace-Beltrami operator on the manifold. Spectral graph theory is a fast developing field in modern discrete mathematics with important applications in computer science, chemistry and operational research. By

merging combinatorial techniques with algebraic and analytical methods it creates new approaches to hard discrete problems and gives new insights in classical linear algebra.

Most spectral methods have a basic framework in common. This basically relies on constructing a matrix M that represents a discrete operator based on the structure of the input graph . This matrix can be seen as incorporating pairwise relations between graph vertices. The pairwise relations can take into account only vertex connectivity or combine topological and geometric information. Hence, an eigendecomposition of the matrix M is performed, that is, its eigenvalues and eigenvectors are computed. Finally, the eigendecomposition is employed in a problem-dependent manner to obtain a desired solution. In view of this framework, the variations for the different spectral methods arise in how the matrix M is composed and how the eigendecomposition is employed to achieve the result, since eigenvalues, eigenvectors, or eigenspace projections can all be used.

A great number of spectral methods have been proposed in the computing science literature in recent years, appearing in the fields of graph theory, computer vision, machine learning, visualization, graph drawing, high performance computing, and computer graphics. Generally speaking, a spectral method solves a problem by examining or manipulating the eigenvalues, eigenvectors, eigenspace projections, or a combination of these quantities, derived from an appropriately defined linear operator. More specific to the area of geometry processing and analysis, spectral methods have been developed with the intention of solving a diversity of problems including mesh compression, correspondence, parameterization, segmentation, sequencing, smoothing, watermarking, surface reconstruction, and remeshing. As a consequence of these developments, researchers are now faced with an extensive literature related to spectral methods, yet this is a topic that still instigates much interest, and there are still many open problems to be addressed, which provide numerous potential possibilities for further investigation.

In fact the importance of the spectral geometry comes from being a topic

which lies in the area of interaction between physics and mathematics. That is to say eigenvalue problems involving the Laplace operator on manifolds (and related objects such as graphs) have proven a constantly fruitful area of mathematical discovery, and acquire deep connections to number theory, physics, and applied mathematics. The study of graph eigenvalues realizes increasingly rich connections with many areas of mathematics. A particularly important development is the interaction between spectral graph theory and differential geometry. There is an interesting analogy between spectral Riemannian geometry and spectral graph theory. The concepts and methods of spectral geometry bring useful tools and crucial insights to the study of graph eigenvalues, which in turn lead to new directions and results in spectral geometry.

In mathematics, the differential geometry of surfaces deals with smooth surfaces with various additional structures, most often, a Riemannian metric. Surfaces have been extensively studied from various perspectives: extrinsically, relating to their embedding in Euclidean space and intrinsically, reflecting their properties determined solely by the distance within the surface as measured along curves on the surface. One of the fundamental concepts investigated is the curvature, where curvature is an intrinsic property of a surface, independent of its isometric embedding in Euclidean space. Surfaces naturally arise as graphs of functions of a pair of variables, and sometimes appear in parametric form associated to space curves. An important role in their study has been played by Lie groups, namely the symmetry groups of the Euclidean plane, the sphere and the hyperbolic plane. These Lie groups can be used to describe surfaces of constant Gaussian curvature; they also provide an essential ingredient in the modern approach to intrinsic differential geometry through connections. On the other hand extrinsic properties relying on an embedding of a surface in Euclidean space have also been extensively studied.

This is well illustrated by the non-linear Euler-Lagrange equations in the calculus of variations: although Euler developed the one variable equations to understand geodesics, defined independently of an embedding, one of Lagrange's main applications of the two variable equations was to minimal surfaces, a con-

cept that can only be defined in terms of an embedding.

To this point we turn our attention to a branch in mathematics which recently developed a "calculus" on graphs that allows graph theory to have new connections to analysis (Friedman & Tillich, 2004a). One key point in this "calculus on graphs" is that, for what appears to be the first time, "non-linear" functions (functions that are not edgewise linear) become important; in previous approaches that unify graph theory and analysis (e.g. (Friedman, 1993) and the references therein) only linear functions are ultimately used. The use of non-linear functions allows many proofs and ideas to carry over more simply from analysis to graphs and vice versa.

Another benefit of the calculus on graphs is that it enables more analysis techniques to carry over to graphs and vice versa in a very direct and simple fashion; less intuition is obscured in technicalities that are particular to analysis or graphs. In this Calculus point of view, a large number of well known results in graph theory such as results on the eigenvalues of the Laplacian can be viewed as gradient inequalities. The proposed graph calculus gives rise to many new partial differential equations on graphs, particularly a new Laplacian based wave equation . It is also allows most techniques for the non-linear p-Laplacian in analysis to be easily carried over to graph theory.

A recently discovered branch of differential geometry, known as "Generalized Geometry", has received a reasonable amount of interest due to the emergence of several connections with areas of Mathematical Physics. The theory is also of interest because the different geometrical structures are often generalizations of more familiar geometries. Generalized geometries commence to play a progressively more significant role, in spite of the fact that one initial starting point for its formulation is a metric target manifold. The emerging picture is that area metric manifolds are generalized geometries. An area metric may be defined as a fourth rank tensor field which allows to assign a measure to two-dimensional tangent areas, in close analogy to the way a metric assigns a measure to tangent vectors. In physics, "Generalized Geometry" is adapted to the physical motion of string-like particles in the same way that traditional geometry is adapted to the

physical motion of point-like particles. More general Generalized Geometries are useful in connection with higher dimensional objects such as membranes. Actually, in more than three dimensions, area metric geometry is a true generalization of metric geometry. In this sense, we aim to generalize the framework we introduce in Chapter 5 in higher dimensional space. We will introduce the mathematical foundation for that purpose in Appendix I.

1.2 Thesis Goals

The ultimate goal of this thesis is to develop a framework for graph characterization by combining the methods from spectral graph theory and manifold learning theory and to explore whether they can provide a stable and robust graph representation. To achieve this we focus on;

- Embedding the nodes of graphs as points in a Manifold embedded into a Euclidean or pseudo Euclidean space. After the embedding we study the geometry of the Manifold to learn the graph properties.
- Extracting stable and robust geometric invariants that can be used for characterizing the graphs aiming at preserving the local manifold structure.
- Representing a graph by a set of curvatures associated with its edges, nodes or triangular faces. The curvature is an intrinsic property of a manifold, independent of its isometric embedding in Euclidean or pseudo Euclidean space. The graph representations will be constructed based on spectral analysis of the graph for the purposes of efficient graph matching and clustering.
- Using the manifold regularization to overcome the effects of noise while preserving the geometrical structures existing in the initial data.
- Constructing the Wave kernel embedding matrix which is mainly based on the edge based Laplacian. The concept comes from a recently developed calculus on graphs. With the embedding matrix in hand, the nodes of

the graph can be embedded in pseudo-Riemannian Manifold into pseudo-Euclidean space.

1.3 Thesis outline

Having described the overall goals of the thesis in this Chapter, we proceed to give a brief overview for the thesis structure:

- Chapter 2 reviews the relevant literature and background for spectral graph theory and its applications in pattern recognition as well as a survey for methods from the manifold learning theory and ending up with a brief survey for graph calculus.
- Chapter 3 explores how to use the heat kernel for the purpose of characterizing graphs in a geometric manner. The new graph representations use sets of curvatures defined either over the edges or triangular faces of the graphs under consideration.
- Chapter 4 presents a process for regularizing the curvature attributes associated with the geometric embedding of graphs.
- Chapter 5 describes a new approach for embedding graphs on pseudo-Riemannian manifolds based on the wave kernel.
- The final chapter 6 gives conclusions and focuses on the advantages and shortcomings of the methods described through the thesis. We also point out some promising directions for future research.

CHAPTER 2

Literature Review

2.1 Introduction

Since our aim in the thesis is to develop methods for graph characterization by combining spectral graph theory and manifold learning theory, this chapter is dedicated for reviewing the relevant literature. We commence with the spectral graph theory and its applications in computer vision and pattern recognition in Section 2.2. Followed by Section 2.3, which is a survey for methods from the manifold learning theory. Finally, Section 2.4 is devoted for the literature relevant to a recently developed graph calculus.

2.2 Overview of the spectral approach in pattern recognition

Spectral graph theory (Biggs, 1993; Chung, 1997; Sachs, Cvetkovic & Doob, 1980) is that branch of mathematics which aims to utilize the eigenvalues and eigenvectors of the adjacency matrix or the closely related Laplacian matrix to

characterize the structural properties of graphs. The use of spectral graph theory in Computer Vision and Pattern Recognition is a recent development, and has proved to be a powerful tool for image segmentation and object recognition (Horaud & Sossa, 1995; Sengupta & Boyer, 1998; Shokoufandeh, Dickinson, Siddiqi & Zucker, 1999). Actually, the classification of shape or object can be posed as clustering spectral features extracted from graph representation that abstract the structure (Luo, Wilson & Hancock, 2003; Wilson, Hancock & Luo, 2005). One of the most important matrices in spectral graph theory is the adjacency matrix. Where, representing graphs in terms of their adjacency matrices open up the possibility of using tools from linear algebra to study the properties of graphs. The earliest literature on algebraic graph theory can be traced back to that of Collatz and Sinogowitz (Collatz & Sinogowitz, 1957). Since then, a large body of literature has emerged aimed at exploiting the relationship between the spectral and structural properties of a graph. This literature is well documented in several surveys including (Biggs, 1993; Doob, Sachs & Cvetkovic', 1995; Chung, 1997; Mohar, 1997). The set of eigenvalues of the adjacency or the Laplacian matrix of a graph is referred to as the graph spectrum (Biggs, 1993). The spectrum can be computed quickly and it conveys many important properties of a graph. Furthermore, the isomorphism of two graphs can also be determined by their spectra. If the eigenvalues of the adjacency matrices of the two graphs are not equal, then the graphs will not be isomorphic (although the converse does not apply).

Although the adjacency matrix and its spectrum have been studied for understanding the structure of graphs, their properties are mostly understood for specific graphs (such as regular graphs, symmetric graphs, random graphs and line graphs). In order to bring spectral methods to a more general family of graphs, many researchers seek answers from the link between spectral graph theory and differential geometry (Fiedler, 1993; Chung, 1997). A study of the Laplacian matrix as well as its eigenspectrum can be found in (Chung, 1997; Merris & Grone, 1994; Grone, 1991; Merris, 1994, 1995; Mohar, 1991, 1992). Therefore, the starting point for most graph spectral methods is the Laplacian matrix, i.e. the degree matrix minus the adjacency matrix. The Laplacian matrix is posi-

tive semi-definite and the multiplicity of the zero eigenvalue gives the number of connected components of the graph. The eigenvector associated with the second smallest eigenvalue can be used to bipartition the nodes of the graph into disjoint subsets of nodes, and this is the basis of a number of data clustering algorithms (Shi & Malik, 2000). Recently, the spectrum of the Laplacian matrix has been used to embed the nodes of a graph into a vector space (He, Yan, Hu, Niyogi & Zhang, 2005). In this space the clustering of nodes can be found using standard clustering techniques such as k-means.

Closely related to the Laplacian spectrum is the heat equation. According to the heat equation the heat kernel can be found by exponentiating the spectrum of the Laplacian matrix with time. The heat kernel is a compact representation of the path length distribution on a graph, and determines information diffusion along edges of the graph with time. An embedding of the nodes of a graph into a vector space may also be performed using a Young-Householder decomposition (Xiao & Hancock, 2004) of the heat kernel. This embedding offers the advantage that the time parameter can be used to control the condensation of clusters. If the nodes of a graph are viewed as residing on a manifold, the Laplacian matrix may be regarded as the discrete approximation to the Laplacian-Beltrami curvature operator for the manifold. In the mathematics literature the study of the eigenvalues and eigenvectors of the Laplace-Beltrami operator is referred to as spectral geometry. In the manifold learning literature (Hein, Audibert & Von Luxburg, 2005) techniques from spectral geometry have recently been used to analyze the properties of the Laplacian embedding.

One of the most important tasks in high-level vision is pattern matching, since it provides a means by which abstract pictorial descriptions can be matched to one another. Recently, there have been many attempts to use spectral graph theory both in graph matching and in point-set matching problems. In (Umeyama, 1988), Umeyama provided one of the earliest attempt where he developed a singular value decomposition method to find the permutation matrix between the adjacency matrices of the two graphs to be matched. His method commences by performing singular value decomposition on the adjacency matrices of the two

graphs separately. The permutation matrix is found by taking the outer products of the eigenvector matrices for the adjacency matrices of the graphs being matched. The method can cope with both weighted and unweighted graphs, but it cannot handle graphs which have a different number of nodes. In (Scott & Longuett-Higgins, 1991), Scott and Longuett-Higgins have shown how to recover correspondence between sets of points by maximizing the inner product of the pairing matrix and the proximity matrix of the two point-sets. An extension of Scott and Longuett-Higgins's idea was introduced by Shapiro and Brady (Shapiro & Brady, 1992), who overcome the shortcoming of their method (which fails to find the correct correspondence when the rotation angle between the point-sets becomes large) by computing the eigenvectors of the proximity matrices of the two point-sets being matched. In fact, both Scott and Longuett-Higgins', Shapiro and Brady's methods can only match point-sets and they cannot be applied directly to graph matching problems. However, there have been many attempts to overcome these limitations. For example, in (Luo & Hancock, 2001) Luo and Hancock have improved Umeyama's method by incorporating the EM algorithm. This allows Umeyama's method to render robustness to the differences in graph size and structural errors. However, the resulting algorithm is time consuming due to its iterative character. Another spectral graph matching method was proposed by Robles-Kelly and Hancock (Robles-Kelly & Hancock, 2002), who aligned the leading eigenvectors of the adjacency matrices of two graphs, where the leading eigenvector corresponds to the steady-state Markov chain. In (Carcassoni & Hancock, 2003), Carcassoni and Hancock provided a method based on Shapiro and Brady's point-set matching algorithm. Where they have shown that by using the EM algorithm, which can incorporate the structure of the point-sets, the confidence of point correspondence can be computed by probabilities using the proximity matrix. Kosinov and Caelli (Kosinov & Caelli, 2002a) have improved Shapiro and Brady's method by allowing for scaling in the eigenspace.

In the rest of this section we will review some problems in computer vision and pattern recognition that have been solved using spectral graph theory;

these incluse spectral graph theory for graph embedding, spectral graph theory for graph matching and spectral graph theory for graph clustering.

2.2.1 Spectral methods for Graph Embedding problems

The aim of graph embedding is to explicitly establish a mapping between graphs and real vectors in order to be able to operate in the associated space, creating some simpler graph based tasks such as matching and clustering. In the literature, different graph embedding methods have been proposed so far. Quite a number of these methods are based on the spectral graph theory. Others take advantage of similarity measures to perform the embedding tasks. Spectral graph theory is based on the analysis of the spectral decomposition of the adjacency matrix or the Laplacian matrix of a graph. The spectrum of these matrices suggests interesting properties about the structure and the topology of the graph. This is why it has been used as the basis for converting graphs into vectors. Spectral graph embedding plays an important role in dimensionality reduction. It typically commences with an affinity matrix computed from the distances between pairs of data points. This data representation is characterized using eigenspectrum affinity matrix; often use one or just a few eigenvectors. For example, principle component analysis (PCA) (Hotelling, 1933) and kernel principle component analysis (KPCA) (Scholkopf, Smola & K.-R. Muller, 1998) use the leading eigenvectors of the covariance matrix to determine the projection directions with the maximal variance. Multidimensional scaling (MDS) (Kruskal & Wish, 1978) uses the eigenvectors of pairwise distance matrix to find an embedding of the data that minimize the stress. As an extension, the isometric feature mapping (ISOMAP) (Tenenbaum, de Silva & Langford, 2000) employs MDS to preserve the geodesic distances between data points located on a manifold. Locally linear embedding (LLE) (Roweis & Saul, 2000) maps the input data to a lower dimensional space in a manner that preserves the local neighbourhood. Similar ideas used in the study of Saerens et al (Saerens, Fouss, Yen & Dupont, 2004).

Based on the adjacency matrix of a graph, a relatively early approach was proposed in (Luo et al., 2003). Where the authors used some spectral features ex-

tracted from the adjacency matrix of a graph to construct a vector representation for the graphs. Hence, embedding these vectors into eigenspaces with the use of the eigenvectors of the covariance matrix of the vectors. Finally, they applied this approach in some graph clustering experiments. In (Wilson et al., 2005), a similar approach has been presented where the coefficients of some symmetric polynomials constructed from the spectral features of the Laplacian matrix were used to represent the graphs into a vectorial form. On a recent paper (Robles-Kelly & Hancock, 2007), the idea was to embed the nodes of a graph into a metric space and view the graph edge set as geodesics between pairs of points in a Riemannian manifold. This was done using the Laplace-Beltrami operator and the Laplacian matrix. Then, the problem of matching the nodes of a pair of graphs is viewed as the alignment of the embedded point sets. In another work (Shokoufandeh, Macrini, Dickinson, Siddiqi & Zucker, 2005) the goal was to obtain a signature to describe shapes using the recursive spectral decomposition of the shock graph representing the skeleton of the shape. A different approach (Xiao & Hancock, 2004) is based on applying metric multidimensional scaling techniques (MDS) to a matrix of shortest geodesic distances between nodes of the graph. The embedding is then used for graph matching. For the special case of trees, an embedding has been defined using the super-tree of a set of sample trees (Torsello & Hancock, 2007). Then, each tree is embedded in a vector where each component is related to one of the nodes of the super-tree and it only has a value different from zero if the node belongs to the specific tree. The method is used in shape analysis using shock trees extracted from the skeletons of 2D shapes. Random walks, and particularly quantum walks have also been used to embed the nodes of a graph in a vector space (Emms, Wilson & Hancock, 2007). In this case the embedding is based on the commute time, the expected time for the walk to travel between two nodes. Another class of graph embedding procedures is based on the selection of some prototypes and the computation of the graph edit distance between the graph and the set of prototypes. This approach was first presented in (Riesen, Neuhaus & Bunke, 2007), and it relies on the work proposed in (Pekalaska, Duin & Paclik, 2006). The basic intuition of this work

is that the description of the regularities in observations of classes and objects is the basis to perform pattern classification. Thus, from the selection of concrete prototypes, each point is embedded into a vector space by taking its distance to all these prototypes. Assuming these prototypes have been appropriately chosen, each class will form a compact zone in the vector space. An extension to map string representations into vector spaces using a similar approach was later proposed in (Spillmann, M., Bunke, Pekalaska & Duin, 2006).

2.2.2 Spectral methods for graph matching problems

Starting from the early seventies of the last century, graph-based techniques have been proposed as a powerful tool for pattern representation and classification in structural Pattern Recognition. After the initial interest induced by the interesting invariance properties of this data structure, graphs have been practically left unused for a lengthy period of time. Recently, the use of graphs in Pattern Recognition is obtaining a growing attention again. This is perhaps due to the fact that the computational cost of the graph-based algorithms, although still high in many cases, is now becoming compatible with the computational power of new computer generations. In the literature there have been a number of attempts to use spectral properties for graph matching. Among the early works on spectral methods is the paper by Umeyama (Umeyama, 1988) the algorithm introduced in this work was for the weighted isomorphism between two graphs, an important restriction here is that the proposed matching method requires a pair of graphs of the same size and the matching matrix must be a permutation matrix (So all the nodes must participate to the matching). In this paper Umeyama used the eigendecomposition of adjacency matrices of the graphs to deduce a simple expression of the orthogonal matrix that optimizes the objective function, under the assumption that the graphs are isomorphic. Unfortunately (as the author suggests), if the graphs are far different from the isomorphic cases this method can produce a very poor results. A more recent paper the one introduced by Xu and King in (Xu & King, 2001), introduces a solution to the weighted isomorphism problem that combines the use of eigenvalues-eigenvectors with continuous op-

timization techniques. The authors reported that the proposed approach is faster than Umeyama, especially with the large scale databases; moreover, it is planner rotationally invariant. As well , in 2001, Carcassoni and Hancock (Carcassoni & Hancock, 2001), proposed a method that is based on the spectral features to define clusters of nodes that are able to be matched together in the optimal correspondence; this method uses hierarchical matching by first finding a correspondence between clusters and then between nodes in the cluster. This method could do with graphs in different sizes.

Using the scenario of combining a spectral approach with the idea of clustering, another method was introduced in 2002 by Kosinov and Caelli (Kosinov & Caelli, 2002b), in which , a vector space called the graph eigenspace, is defined using the eigenvectors of the adjacency matrix, and the nodes are projected onto points in the space. Using the authors words, the two most important properties of their approach are, first its ability to match graphs of considerably different sizes, and second, its power to discover correspondence relationships among subgraphs and groups of vertices.

Another method was introduced by Shokoufandeh and Dickinson in 2001 (Shokoufandeh & Dickinson, 2001), which to some extent can be related to spectral techniques, in their work the authors used the eigenvalue characterization of a directed acyclic graph to map its topological structure into a low-dimensional vector space. As pointed out by the authors the algorithm should work well on graphs with any rooted hierarchical structure, whether directed acyclic graph or rooted tree, but it does not give any guarantee of optimality.

2.2.3 Spectral methods for Graph Clustering problem

Clustering is one of the most widely used techniques for exploring data structures and has found increasing support and applications in many areas ranging from statistics, computer science, biology to social sciences or psychology (e.g. (Jain, Murty & Flynn, 1999; Shi & Malik, 2000; Xu & Wunsch, 2005; Newman, Watts & Strogatz, 2002)). Almost in every scientific field dealing with practical data, people attempt to get a first impression on the data by trying to identify groups

of "similar behavior" in their data. The aim of clustering methods is to group patterns on the basis of a similarity (or dissimilarity) criteria where groups (or clusters) are set of similar patterns. Essential aspects in clustering are pattern representation and the similarity measure. Each pattern is represented by a set of features of the system under study. Once a representation is fixed it is possible to choose an appropriate similarity measure among patterns. The most popular dissimilarity measure for metric representations is the distance, for instance the Euclidean one (Duda & Hart, 1973).

In this section we aim to focus on spectral clustering methods, which has become quite popular over the last few years, it is very simple to implement and can be solved efficiently by standard linear algebra methods. Spectral clustering is an approach able to produce nonlinear separating hypersurfaces between clusters, arising from concepts in spectral graph theory (Chung, 1997) and the clustering problem is configured as a graph cut problem where an appropriate objective function has to be optimized. Using information obtained from the eigenvalues and eigenvectors of the adjacency matrices, Spectral clustering methods create partitioning of graphs. The basic idea is to construct, from the initial data set, a weighted graph. Each node represents a pattern and each weighted edge simply takes into account the similarity between two patterns. In this framework the clustering problem can be seen as a graph cut problem, which can be tackled by means of spectral graph theory. The core of this theory is the singular values decomposition of the Laplacian matrix of the weighted graph obtained from data, which is related to its cut. And even for large data sets, spectral clustering can be implemented efficiently using a sparse similarity graph (Verma & Meila, 2005).

A comparison of some spectral clustering methods has been recently proposed in (Verma & Meila, 2005; Luxburg, 2007), while there are some theoretical works on the capabilities and convergence properties of spectral methods for clustering (Kannan, Vempala & Vetta, 2000; Luxburg, Belkin & Bousquet, 2004; Luxburg, Bousquet & Belkin, 2005; Zha, He, Ding, Gu & Simon, 2001). The Spectral methods have been applied in clustering of artificial data (Ng, Jordan & Weiss, 2002; Rahimi & Recht, 2004), in image segmentation (Barreno,

2004; Meila & Shi, 2000; Shi & Malik, 2000; Srivastava, 2004), in bioinformatics (Cristianini, Taylor & Kandola, 2001), in social network analysis (Newman et al., 2002), and in co-clustering problems of words and documents (Dhillon, 2001) and genes and conditions (Kluger, Basri, Chang & Gerstein, 2003). In (Kulis, Basu, Dhillon & Mooney, 2005), a semi-supervised spectral approach to bioinformatics and handwritten character recognition has been proposed. And the protein sequence clustering problem has been faced using spectral techniques in (Paccanaro & Saqi, 2003).

2.3 Manifold Learning

The focus on manifold learning is mainly motivated by the need to process more complex features that are naturally represented as points on a manifold, hidden in high-dimensional spaces such as images. Quite often there is a need to quantify various phenomena which are obvious for a human observer, but difficult to describe in mathematical terms. Texture, shape and many other aspects of data need to be quantified and compared, and the mathematical theory of smooth manifolds is a natural approach for many such problems.

Learning a manifold of perceptual observation is difficult where these observations usually exhibit significant nonlinear structure. Classical techniques for manifold learning, such as principle component analysis (PCA) (Hotelling, 1933; Jolliffe, 1986) and metric multidimensional scaling (MDS) (Jolliffe, 1986) are designed to operate when the manifold is embedded linearly or almost linearly in the ambient space. Both of these methods are spectral ones, i.e., methods based on eigenvalue decomposition of either the covariance matrix (for PCA) or the Gram matrix (for MDS) of the input data. For data sampled from general nonlinear manifolds, however, these linear methods do not give satisfactory answers. In recent times, a number of new spectral methods have been developed to discover the nonlinear structure of the manifold such as Isomap (Tenenbaum et al., 2000), locally linear embedding (LLE) (Roweis & Saul, 2000), Laplacian eigenmaps (Belkin & Niyogi, 2003), Hessian LLE (Donoho & Grimes, 2003),

maximum variance unfolding (MVU) (Weinberger & Saul, 2006), local tangent space alignment (Zhang & Zha, 2004) and geodesic nullspace analysis (Brand, 2004). In (Saul, Weinberger, Sha, Ham & Lee, 2005) and (Burges, 2005), one can find a tremendous summary of these methods. As mentioned in (Saul et al., 2005), even though these new methods have a similar computational structure, they are based on rather different geometric intuitions and intermediate computations. For instance, Isomap tries to preserve the global pairwise distances of the input data as measured along the low dimensional manifold (geodesic distances); LLE and Laplacian eigenmaps try to preserve certain local geometric relationships of the data; MVU, on the other hand, preserves local distances but maximize a global objective (the total variance).

Although, these nonlinear methods do yield impressive results on some data sets and some real applications, their nonlinear property makes them computationally expensive. For instance, Isomap and MVU construct a dense matrix and use its top eigenvectors (eigenvectors associated with the largest eigenvalues) in producing the low dimensional representations, while LLE, Laplacian eigenmaps, and Hessian LLE construct a sparse matrix and use its bottom eigenvectors (eigenvectors associated with the smallest eigenvalues). In addition, methods using dense matrices (Gram matrix) can often distinguish the intrinsic dimension by a tellable gap between a few top eigenvalues and the rest of the spectra, but methods using sparse matrices (e.g., Laplacian) do not yield such an estimate since their bottom eigenvalues are usually closely located. In the latter case, an additional step of estimating the intrinsic dimensionality is needed beforehand; see, e.g., (Costa & Hero, 2004) and references therein.

As stated in (Saul et al., 2005), each of these spectral methods for dimensionality reduction has its own advantages and disadvantages and each can be preferable for different classes of data sets. In (Ham, Lee, Mika & Scholkopf, 2004), a kernel view of these algorithms was given for a better understanding of the connections between these methods, interpreting each of them on specially constructed kernel matrices.

Although several nonlinear techniques have been proposed to discover the

nonlinear structure of the manifold, their nonlinear property makes them computationally expensive. Moreover, most of these methods do not explicitly consider the structure of the manifold on which the data may possibly reside.

2.4 Calculus on graphs

In (Friedman & Tillich, 2004a), Friedman and Tillich developed a Calculus on Graphs that allows graph theory to have new connections to analysis. This framework gives rise to many new partial differential equations on graphs, in particular a new wave equation based on an edge-based Laplacian. Such wave equation gives rise to partial improvements on several concepts in graph theory and in analysis (e.g.(Chung, Faber & Manteuffel, 1994; Chung, Grigor'yan & Yau, 1996, 1997; Bobkov & Ledoux, 1997)). A feature point in this graph calculus is that the "non-linear" functions, those which are not edgewise linear become more applicable; while in previous approaches that unify graph theory and analysis (e.g. (Friedman, 1993) and the references therein) only linear functions are used. The use of non-linear functions allows many proofs and ideas to carry over more simply from analysis to graphs and vice versa. Moreover, some new variants and simpler proofs of inequalities known in graph theory as in (Diaconis & Saloff-Coste, 1996; Saloff-Cost, 1997; Coulhon, 1992, 1996a; Bakry, Coulhon, Ledoux & Saloff-Coste, 1995; Coulhon & Grigor'yan, 1997; Coulhon, 1996b; Chung & Yau, 1995) were deduced.

2.5 Conclusion

Reviewing the relevant literature and background for the work to be done in this thesis, we may draw several conclusions:

First, due to the remarkable development of using computers, pattern recognition becomes an enabling technology in more and more applications in the practical world. Hence, improving pattern recognition algorithms become a key to mathematics and computer science research. With the Spectral Graph The-

ory becoming very popular in many areas such as computer science, chemistry, network design and coding theory; graph-based algorithms have received a great attention recently, yet this is a topic that still attracts much interest, and there are still many open problems to be addressed, which provide numerous potential possibilities for further investigation. And for the importance of the heat kernel in spectral graph theory, it is convenient to explore how to use the heat kernel to characterize graphs in a geometric manner.

Secondly, classical techniques for manifold learning are designed to operate when the manifold is embedded linearly or almost linearly in the ambient space. And even though several nonlinear techniques have been proposed to discover the nonlinear structure of the manifold, their nonlinear property makes them computationally expensive. Moreover, most of these methods either consider some global structures of the manifold on which the data may possibly reside or preserve a local feature while regarding a global structure of the manifold. We aim to use the curvature as it is an intrinsic property of a manifold, independent of its isometric embedding in Euclidean or pseudo Euclidean space, to characterize the graph for the purposes of efficient graph matching and clustering.

Thirdly, as surfaces naturally arise as graphs of functions, we may use concepts deduced from a recently developed calculus on graphs in an attempt to connect graph theory to analysis. This graph calculus allows most techniques for the non-linear operators in analysis to be easily carried over to graph theory.

CHAPTER 3

Heat Kernel Embedding

3.1 Introduction

Kernel-based methods provide a powerful framework for application areas ranging from neural networks and pattern recognition to machine learning and data mining. In pattern recognition they have led to the development of a number of methods including kernel-based Principal Component Analysis (KPCA) (Scholkopf et al., 1998). Kernel methods motivate algorithms that can act on general types of data such as vectors, strings or text and look for general types of relations such as clusters, classifications or regressions. One of the most important kernel-based methods is the heat kernel which is found by solving the diffusion equation for the discrete structure under study. The heat kernel associated with a second-order partial differential equation in a Euclidean space, is an important analytical tool in physics and has been used in many other areas including spectral graph theory (Chung, 1997). Recent work by Smola and Kondor (Smola & Kondor, 2003a) has shown how kernels can be used to graphs, also number of alternatives has been suggested and compared. A kernel function

is defined, which implicitly maps each graph into a high-dimensional feature space, hence clustering may be performed in a space in which the classes are more easily separated using standard clustering techniques such as $k-$means. An embedding of the nodes of a graph into a vector space may also be performed using a Young-Householder decomposition (Young & Householder, 1938; Xiao & Hancock, 2004) of the heat kernel. This embedding offers the advantage that the time parameter can be used to control the condensation of clusters. If the nodes of a graph are viewed as residing on a manifold, the Laplacian matrix may be regarded as the discrete approximation to the Laplacian-Beltrami curvature operator for the manifold. In the mathematics literature the study of the eigenvalues and eigenvectors of the Laplace-Beltrami operator is referred to as spectral geometry. In the manifold learning literature (Hein et al., 2005) techniques from spectral geometry have recently been used to analyze the properties of the Laplacian embedding.

The Laplacian matrix can be interpreted as a matrix representation of the graph and its spectrum has been widely studied in spectral graph theory (Chung, 1997) and has proved to be a versatile mathematical tool that can be put to many practical uses including routing (Atkins, Boman & Hendrickson, 1998), indexing (Shokoufandeh et al., 1999), clustering (Shi & Malik, 2000) and graph-matching (Umeyama, 1988; Luo & Hancock, 2001). One of the most important properties of the Laplacian spectrum is its close relationship with the heat equation. The heat equation can be used to specify the flow of information with time across a network or a manifold (Yau & Schoen, 1988). According to the heat-equation the time derivative of the kernel is determined by the graph Laplacian. The solution to the heat equation is obtained by exponentiating the Laplacian eigensystem over time. Since the heat kernel encapsulates the way in which information flows through the edges of the graph over time, it is closely related to the path length distribution on the graph. Recently, Lebanon and Lafferty (Lebanon & Lafferty, 2004) have shown how they used the heat kernel to construct statistical manifolds that can be used for inference and learning tasks. Moreover, in (Xiao & Hancock, 2006), the authors have explored how a number of different invariants

that can be computed from the heat kernel can be used for graph clustering. Colin de Verdiere has shown how to compute geodesic invariants from the Laplacian spectrum (de Verdi‘ere, 1998). In fact, a graph can be viewed as residing on a manifold whose pattern of geodesic distances is characterized by the heat kernel. Differential invariants can be computed from the heat kernel, and these in turn are related to the Laplacian eigensystem. This field of study is sometimes referred to as spectral geometry (Gilkey, 1984; Yau & Schoen, 1988). One of the most interesting recent developments in this area is to establish a link between graph-spectra and the geometry of the underlying manifold (Grigor'yan, 2001, 2006; Zhu, Kandola, Ghahramani & Lafferty, 2004; Barlow, 1998; Smola & Kondor, 2003a). In (Grigor'yan, 2006, 2001; Barlow, 1998), a considerable insight can be achieved through the analysis of the heat kernel of the graph. There are a number of different invariants that can be computed from the heat-kernel. Asymptotically for small time, the trace of the heat kernel (Chung, 1997) (or the sum of the Laplacian eigenvalues exponentiated with time) can be expanded as a rational polynomial in time, and the co-efficients of the leading terms in the series are directly related to the geometry of the manifold. For instance, the leading co-efficient is the volume of the manifold, the second co-efficient is related to the Euler characteristic, and the third co-efficient to the Ricci curvature. The aim in this chapter is to investigate whether the heat kernel can be used to provide a geometric characterization of graphs that can be used for the purposes of graph-clustering. This is of course a problem that can be addressed directly by using the spectral geometry of the combinatorial Laplacian. However, there are two major obstacles. First, the results delivered by spectral geometry are interesting, they are applied under the assumption that the graph Laplacian converges to the corresponding continuous Laplace operator provided that the graph is sufficiently large. Second, the calculations involved are complicated and the resulting expressions are not very elegant. Hence, we adopt a more pragmatic approach in this chapter where we aim to characterize the geometry of point distribution based on embeddings derived from the heat-kernel.

The method involves performing a Young-Householder decomposition of the

heat-kernel to recover the matrix of embedding co-ordinates. In other words, we perform kernel principal components analysis on the heat kernel to map nodes of the graph to points in a manifold. We provide an analysis which shows how the eigenvalues and eigenvectors of the covariance matrix for the point distribution resulting from the kernel mapping are related to those of the Laplacian. With the embeddings to hand, we develop a graph characterization based on differential geometry. To do so we compute the sectional curvatures associated with the edges of the graph, making use of the fact that the sectional curvature is determined by the difference between the geodesic and Euclidean distances. taking this analysis one step further, we use the Gauss-Bonnet theorem to compute the Gaussian curvatures associated with triangular faces of the graph. We characterize graphs using sets of curvatures, defined either on the edges or the faces. We explore whether these characterizations can be used for the purposes of graph matching. To this end, we compute the similarities of the sets using robust variants of the Hausdorff distance which allows us to compute the similarity of different graphs without knowing the correspondences between edges or faces. The rest of this chapter is organized as follows: In Section 3.2 we provide some background on the heat-kernel and its relationship with the Laplacian spectrum. In Section 3.3 we develop our geometric characterization of graphs. In Section 3.4 we experiment with the method on three different Real-world databases, namely the York model house, the COIL-20 and the TOY image sequences.

3.2 Heat Kernels on Graphs

In this section, we give a brief introduction on the graph heat kernel. We commence in Section 3.2.1 with some important matrices related to the graph under study, followed in Section 3.2.2 by deducing the heat kernel from the heat equation. The relation between the heat kernel and the path length distribution on the graph is given in Section 3.2.3. Then we show in Section 3.2.4 how it can be used to embed the nodes of a graph in a vector space using the Young-Householder decomposition. Finally, in Section 3.2.5, we provide an analysis which reveals

the relationship between the eigenvalues and eigenvectors of the heat-kernel and those of the covariance matrix for the point distribution resulting from the embedding.

3.2.1 Preliminaries

To commence, consider an undirected unweighted graph denoted by $G = (V, E)$ where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. The elements of the adjacency matrix A of the graph G is defined by:

$$A(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

To construct the Laplacian matrix we first establish a diagonal degree matrix D, whose elements are given by the degree of the nodes, i.e. $D(u,u) = deg(u) = \sum_{v \in V} A(u,v)$. From the degree matrix and the adjacency matrix we construct the Laplacian matrix $L = D - A$, i.e. the degree matrix minus the adjacency matrix,

$$L(u,v) = \begin{cases} deg(v) & \text{if } u = v \\ -1 & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

The normalized Laplacian $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ has elements

$$\hat{L}(u,v) = \begin{cases} 1 & \text{if } u = v \text{ and } d_v \neq 0 \\ -\frac{1}{\sqrt{deg(u)deg(v)}} & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

The spectral decomposition of the normalized Laplacian matrix is $\hat{L} = \Phi \Lambda \Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues $(\lambda_1 < \lambda_2 < ... < \lambda_{|V|})$ as elements and $\Phi = (\phi_1 | \phi_2 | .... | \phi_{|V|})$ is the matrix

with the ordered eigenvectors as columns. Since $\hat{L}$ is symmetric and positive semi-definite, the eigenvalues of the normalized Laplacian are all non-negative. The multiplicity of the zero eigenvalue is the number of isolated cliques in the graph. For a connected graph, the multiplicity of the zero eigenvalue is one. The eigenvector associated with the smallest non-zero eigenvalue is referred to as the Fiedler-vector (Chung, 1997). In many practical situations, it has been shown theoretically that using the Normalized Laplacian leads to more robust semi definite Laplacian (Luxburg et al., 2004). Hence, in our work through this thesis we are using the Normalized Laplacian matrix, $\hat{L}$.

3.2.2 Heat Equation

Here we are interested in the heat equation associated with the Laplacian, which is given by.

$$\frac{\partial h_t}{\partial t} = -\hat{L}h_t \tag{3.4}$$

where h_t is the heat kernel and t is time, and its partial derivative $\frac{\partial h_t}{\partial t}$is to be computed by taking the derivative of each element of h_t . The heat kernel is the fundamental solution of the heat equation. It can be viewed as describing the flow of information across the edges of the graph with time. The rate of flow is determined by the Laplacian of the graph. The solution to the heat equation at time t, can be computed through the heat kernel h_t

$$h_t = e^{-t\hat{L}} \tag{3.5}$$

The matrix exponential $(e^{-t\hat{L}})$, is a matrix function on the square matrix $\hat{L}$ analogous to the ordinary exponential function. One way to approximate $(e^{-t\hat{L}})$, is to use the eigen-decomposition of the Laplacian $\hat{L}$. From (Chung, 1997) we can proceed to compute the heat kernel on a graph by exponentiating the Laplacian eigenspectrum, i.e.

$$h_t = \Phi \exp[-\Lambda t]\Phi^T = \exp[\hat{L}t] \tag{3.6}$$

The heat kernel is a $|V| \times |V|$ matrix. For the nodes u and v of the graph G the heat kernel element is

$$h_t(u,v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i\phi_i^T = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v) \tag{3.7}$$

When t tends to zero, then $h_t \simeq I - \hat{L}t$, i.e. the kernel depends on the local connectivity structure or topology of the graph (Xiao, Wilson & Hancock, 2005). If, on the other hand, t is large, then

$$h_t \simeq I - \exp[-\lambda_2 t]\phi_2\phi_2^T$$

where λ_2 is the smallest non-zero eigenvalue and ϕ_2 is the associated eigenvector, i.e. the Fiedler vector. Hence, the large time behavior is governed by the global structure of the graph.

3.2.3 Geodesic Distance from the Heat Kernel

It is interesting to note that the heat kernel is also related to the path length distribution on the graph (Xiao et al., 2005). To show this, consider the matrix $P = I - \hat{L}$, where I is the identity matrix. The heat kernel can be rewritten as $h_t = e^{-t(I-P)}$. Hence, we can perform the MacLaurin expansion on the heat kernel to re-express it as a polynomial in t. The result of this expansion is

$$h_t = e^{-t}(I + tP + \frac{(tP)^2}{2!} + \frac{(tP)^3}{3!} + ...) = e^{-t}\sum_{k=0}^{\infty} P^k\frac{t^k}{k!} \tag{3.8}$$

For a connected graph, the matrix P has elements

$$P(u,v) = \begin{cases} 0 & \text{if } u = v \\ \frac{1}{\sqrt{deg(u)deg(v)}} & \text{if } u \neq v \text{ and } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

As a result, we have that

$$P^k(u,v) = \sum_{S_k} \prod_{i=1}^{k} \frac{1}{\sqrt{deg(u_i)deg(u_{i+1})}} \tag{3.10}$$

where the walk S_k is a sequence of vertices $u_0, ..., u_k$ of length k such that $(u_i, u_{i+1}) \in E$. Hence, $P^k(u,v)$ is the sum of weights of all walks of length k joining nodes u and v. In terms of this quantity, the elements of the heat kernel, (Xiao et al., 2005), are given by

$$h_t(u,v) = \exp[-t] \sum_{k=0}^{|V|^2} P^k(u,v) \frac{t^k}{k!} \tag{3.11}$$

We can find a spectral expression for the matrix P^k using the eigendecomposition of the normalized Laplacian. Writing $P^k = (I - \hat{L})^k$ it follows that $P^k = \Phi(I - \Lambda)^k \Phi^T$. The element associated with the nodes u and v is

$$P^k(u,v) = \sum_{i=1}^{|V|} (1 - \lambda_i)^k \phi_i(u) \phi_i(v) \tag{3.12}$$

The geodesic distance between nodes, i.e. the length of the walk on the graph with the smallest number of connecting edges, can be found by searching for the smallest value of k for which $P^k(u,v)$ is non zero, i.e. $d_G(u,v) = floor_k P_k(u,v)$

3.2.4 Heat Kernel Embedding

The nodes of the graph are to be mapped into a vector space using the heat kernel. For that we consider $Y = (y_1|...|y_u|...|Y_{|V|})$ be the $|V| \times |V|$ matrix with the vectors of co-ordinates as columns. The vector of co-ordinates y_u for the node index u is hence the u^{th} column of Y. The co-ordinate matrix is found by performing the Young-Householder decomposition $h_t = Y^T Y$ on the heat-kernel. Since $h_t = \Phi \exp[-\Lambda t]\Phi^T$, $Y = \exp[-\frac{1}{2}\Lambda t]\Phi^T$. Hence, the co-ordinate vector for the node indexed u is

$$y_u = (\exp[-\frac{1}{2}\lambda_1 t]\phi_1(u), \exp[-\frac{1}{2}\lambda_2 t]\phi_2(u), ..., \exp[-\frac{1}{2}\lambda_{|V|} t]\phi_{|V|}(|V|))^T \tag{3.13}$$

The kernel mapping $\mathcal{M} : V \rightarrow \mathcal{R}^{|V|}$, embeds each node on the graph in a vector space $\mathcal{R}^{|V|}$. The heat kernel $h_t = Y^T Y$ can also be viewed as a Gram matrix, i.e. its elements are scalar products of the embedding co-ordinates. Consequently, the kernel mapping of the nodes of the graph is an isometry. The squared Euclidean distance between nodes u and v is given by

$$d_E(u,v)^2 = (y_u - y_v)^T(y_u - y_v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\Big\{\phi_i(u) - \phi_i(v)\Big\}^2 \tag{3.14}$$

Figure 3.1 shows the steps to embed the graph into a manifold.

3.2.5 Point Distribution Statistics

One very simple way to characterize the embedded point-set is to study the properties of the covariance matrix of the point-set generated by the embedding methods. To construct the covariance matrix, we commence by computing the mean coordinate vector (Xiao et al., 2005). The mean co-ordinate vector for the heat

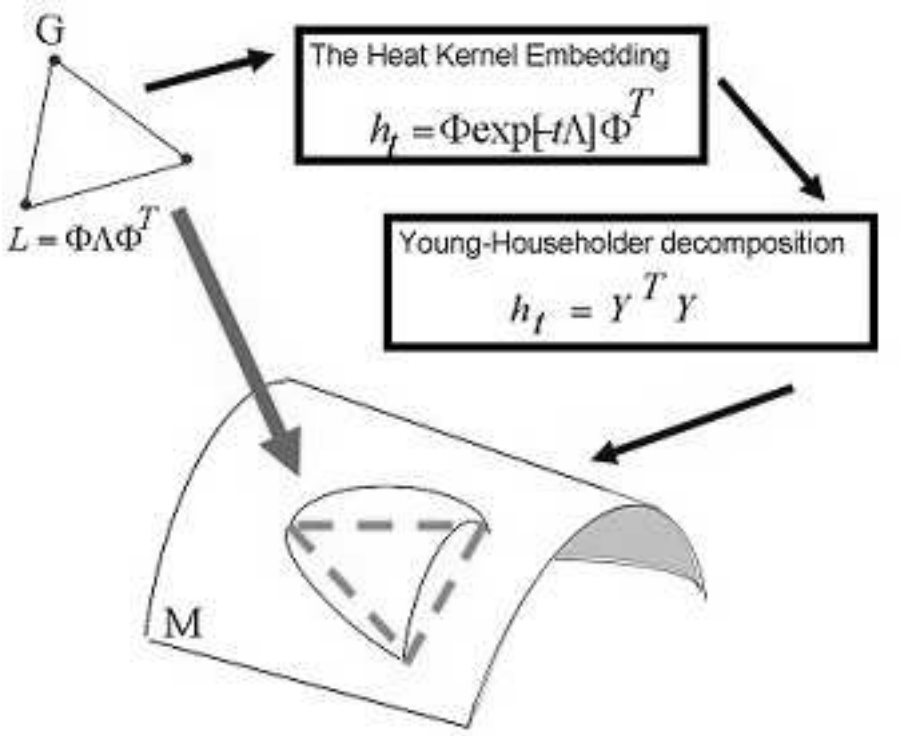


Figure 3.1: Illustration of the geometric embedding of the graph into a manifold.

kernel embedding is

$$\hat{y} = \frac{1}{|V|} Ye = \frac{1}{|V|} \exp[-\frac{1}{2}\Lambda t]\Phi^T e \tag{3.15}$$

where $e = (1, 1, ..., 1)^T$ is the all ones vector of length $|V|$. The matrix of centered co-ordinates is found by subtracting the mean position vector from each of the co-ordinate vectors and is given by

$$Y_C = Y - \frac{1}{|V|} Yee^T = \exp[-\frac{1}{2}\Lambda t]\Phi^T (I - \frac{1}{|V|} ee^T) = \exp[-\frac{1}{2}\Lambda t]\Phi^T M^T \tag{3.16}$$

where $M^T = (I - \frac{1}{|V|} ee^T)$. The covariance matrix for the embedded point-positions is

$$S_Y = \frac{1}{|V|} Y_C Y_C^T = \frac{1}{|V|} \exp[-\frac{1}{2}\Lambda t]\Phi^T M^T M\Phi \exp[-\frac{1}{2}\Lambda t] \tag{3.17}$$

Hence, we can write

$$S_Y = \frac{1}{|V|}C^TC$$

where $C = M\Phi\exp[-\frac{1}{2}\Lambda t]$. To compute the eigenvectors of S_Y we first construct the matrix,

$$CC^T = M\Phi\exp[-\Lambda t]\Phi^TM^T = Mh_tM^T$$

i.e. CC^T has eigenvalue matrix $\Lambda_h = \exp[-\Lambda t]$ and un-normalised eigenvector matrix $U = M\Phi$. As a result the matrix C^TC has normalised eigenvector matrix $\hat{U} = C^TU\Lambda_h^{-\frac{1}{2}}$ and eigenvalue matrix Λ_h. To see this note that

$$(C^TU\Lambda_h^{-\frac{1}{2}})\Lambda_h(C^TU\Lambda_h^{-\frac{1}{2}})^T = C^TUU^TC = C^TC \tag{3.18}$$

Hence C^TC has eigenvector matrix $\Lambda_h = \exp[-\Lambda t]$ and normalised eigenvector matrix

$$\hat{U} = (M\Phi\exp[-\frac{1}{2}\Lambda t])^TM\Phi(\exp[-\Lambda t])^{-\frac{1}{2}} = \exp[-\frac{1}{2}\Lambda t]\Phi^TM^TM\Phi\exp[\frac{1}{2}\Lambda t] \tag{3.19}$$

Finally, it is interesting to note that the projection of the centered co-ordinates onto the eigenvectors of the covariance matrix S_Y is

$$Y_P = \hat{U}^TY_C = \exp[-\frac{1}{2}\Lambda t]\Phi^TM^T = Y_C \tag{3.20}$$

3.3 Geometric Characterisation

Whereas graph embeddings have found widespread use in machine learning and pattern recognition for the purposes of clustering, analyzing and visualization relational data, they have also proved to be useful as a means of graph characterization. When embedding the nodes of a graph on a manifold in a vector space, and to use the geometric properties of the resulting point-set as a graph charac-

teristic. Thinking of curvature as a local measure of geometry, our aim here is to use it to represent local shape information. Actually curvature has only recently been exploited to its full potential, due mainly to the advent of computers. In this section we develop our differential characterisation of graphs using different kinds of curvatures. We commence by showing how the geodesic and Euclidean distances estimated from the spectrum of the Laplacian and the heat kernel embedding can be used to associate a sectional curvature with the edges of a graph. Next, we turn our attention to geodesic triangles formed by the embedding of first order cycles, i.e. triangles of the graph. In that manner we are using Gauss Bonnet theorem, Which states that the sum of interior angles of a geodesic triangle is equal to π plus the total curvature enclosed by the triangle, to compute the Gaussian curvature through the angular excess of the geodesic triangles.

3.3.1 The Sectional Curvature

In this section we show how the Euclidean distance and geodesic distances computed for embedding can be used to compute the sectional curvature associated with edges of the graph. The sectional curvature is determined by the degree to which the geodesic bends away from the Euclidean chord. Hence for a torsionless geodesic on the manifold, the sectional curvature can be estimated easily if the Euclidean and geodesic distances are known. Suppose that the geodesic can be locally approximated by an arc of a circle. Let the geodesic distance between the pair of points u and v be $d_G(u,v)$ and the corresponding Euclidean distance be $d_E(u,v)$. Further let the radius of curvature of the approximating arc be $r_s(u,v)$ and suppose that the tangent vector to the manifold undergoes a change in direction of $2\theta_{u,v}$ as we move along a connecting arc between the two points. We show an illustration of the above in Figure 3.2.
In terms of the angle $\theta_{u,v}$, the geodesic distance, i.e. the distance traversed along the circular arc, is

$$d_G(u,v) = 2r_s(u,v)\theta_{u,v} \tag{3.21}$$

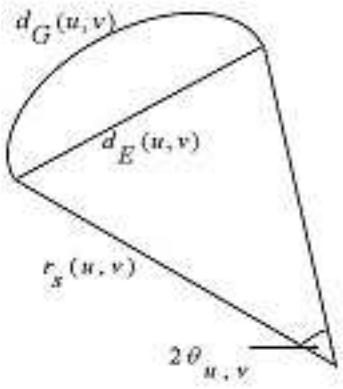


Figure 3.2: Illustration of relationship between the geodesic distance, Euclidean distances and the sectional curvature.

and as a result we find that

$$\theta_{u,v} = d_G(u,v)/2r_s(u,v) \tag{3.22}$$

The Euclidean distance, on the other hand, is given by

$$d_E(u,v) = 2r_s(u,v)\sin\theta_{u,v} \tag{3.23}$$

and can be approximated using the MacLaurin series

$$d_E(u,v) = 2r_s(u,v)\{\theta_{u,v} - \frac{1}{6}\theta_{u,v}^3 + ...\} \tag{3.24}$$

Substituting for $\theta_{u,v}$ obtained from the geodesic distance, we have

$$d_E(u,v) = d_G(u,v) - \frac{d_G(u,v)^3}{24r_s^2(u,v)} \tag{3.25}$$

Solving the above equation for the radius of curvature, the sectional curvature of

the geodesic connecting the nodes u and v is approximately

$$k_s(u,v) = \frac{1}{r_s(u,v)} = \frac{2\sqrt{6}(d_G(u,v) - d_E(u,v))^{\frac{1}{2}}}{d_G(u,v)^{\frac{3}{2}}} \tag{3.26}$$

Since for an edge of the graph $d_G(u,v) = 1$, we have

$$k^2(u,v) = 24(1 - d_E(u,v)) \tag{3.27}$$

3.3.2 The Gaussian Curvature

The Gauss-Bonnet Theorem links the topology and geometry of a surface in an elegant and compact manner. Spivak (Spivak, 1979) and Stillwell (Stillwell, 1974) give accounts of the early history of its development and application. For a smooth compact oriented Riemannian 2-manifold M, let $\triangle_G$ be a triangle on M whose sides are geodesics, i.e. paths of shortest length on the manifold. Further, let α_1, α_2 and α_3 denote the interior angles of the triangle. According to Gauss's theorem, if the Gaussian curvature K (i.e. the product of the maximum and the minimum curvatures at a point on the manifold) is integrated over $\triangle_G$, then

$$\int_{\triangle_G} K dM = \sum_{i=1}^{3} \alpha_i - \pi \tag{3.28}$$

where dM is the Riemannian volume element.

To estimate the Gaussian curvature from the above, we must determine the interior angles α_i of the geodesic triangle. To this end we assume that T is a triangulation of a smooth manifold M and $\triangle_G$ is a geodesic triangle on M with angles $\{\alpha_i\}_{i=1}^3$ and geodesic edge lengths $\{d_{Gi}\}_{i=1}^3$. Moreover we suppose that $\triangle_e$ is the corresponding Euclidean triangle with edge lengths $\{d_{Ei}\}_{i=1}^3$ and interior angles $\{\varphi_i\}_{i=1}^3$. We assume that the geodesic index i is a great arc on a sphere with radius r_i, $i = 1, 2, 3$. Furthermore, we'll treat the geodesic triangles as residing on a hyper$-$sphere with a radius r which is computed by averaging

over the constituent geodesic edges, that is $r = \frac{1}{3}\sum_{i=1}^{3} r_i$. To commence, we compute the area of the geodesic triangle. Here we'll make use of the geometry of the sphere, the area of the spherical triangle is given by

$$A_G = (\sum_{i=1}^{3} \alpha_i - \pi)r^2 \tag{3.29}$$

From (3.29) we can see that

$$\sum_{i=1}^{3} \alpha_i - \pi = \frac{A_G}{r^2} \tag{3.30}$$

Now, considering a small area element on the sphere given in spherical coordinates by $dA_G = r^2 \sin\theta d\theta d\varphi$, the integration of dA bounded by 2θ gives us another formula for computing the area of the geodesic triangle

$$A_G = \int_0^{2\theta}\int_0^{2\theta} r^2 \sin\theta d\theta d\varphi \tag{3.31}$$

$$= r^2(1-\cos 2\theta)(2\theta) \tag{3.32}$$

$$= r^2(2\sin^2\theta)(2\theta) \tag{3.33}$$

$$= (2r\sin\theta)^2(\theta) \tag{3.34}$$

$$\tag{3.35}$$

Substituting from the formulas (3.23) and (3.22) where $d_G = 1$ for an edge of a graph, we get

$$A_G = \frac{1}{2r}d_E^2 \tag{3.36}$$

where d_E^2 is computed from the embedding using (3.14). From (3.28), (3.30) and (3.36), we get the following formula for the Gaussian curvature residing

over the geodesic triangle:

$$\int_{\triangle_G} KdM = \frac{1}{2r^3} d_E^2 \tag{3.37}$$

3.4 Experiments

Representing the graphs using sets of curvatures defined either over the edges (i.e. sectional curvatures) or triangular faces (i.e. Gaussian curvatures) of the graphs under consideration. The sets of curvatures are unordered, i.e. we do not know the correspondences between edges or faces in different graphs, and hence we require a set-based similarity measure to compare graphs in the absence of correspondences. One route is provided by the Hausdorff distance. However, this is known to be sensitive to noise, so we explore median and probabilistic variants of the Hausdorff distance in Section 3.4.1 and Section 3.4.2 respectively.

With the graph distances in hand, we require a means of visualizing the distribution of graphs. The classical Multidimensional Scaling (MDS) (Cox & Cox, 1994) is the method we are using here to embed the data specified in the matrix in a Euclidean space 3.4.3. Finally, the results obtained when experimenting with a real world data are to be given in Section 3.4.5.

3.4.1 Hausdorff distance

The Hausdorff distance provides a means of computing the distance between sets of unordered observations when the correspondences between the individual items are unknown. In its most general setting, the Hausdorff distance is defined between compact sets in a metric space. Given two such sets, we consider for each point in one set is the closest point in the second set. Hausdorff distance is the maximum over all these values. More formally, the classical Hausdorff distance(HD) (Huttenlocher, Klanderman & Rucklidge, 1993) between two finite point sets A and B is given by

$$H(A, B) = \max(h(A, B), h(B, A)) \tag{3.38}$$

where the directed Hausdorff distance from A to B is defined to be

$$h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\| \tag{3.39}$$

and $\|.\|$ is some underlying norm on the points of A and B (e.g., the L2 or Euclidean norm). Dubuisson and Jain (Dubuisson & Jain, 1994) proposed a robust modified Hausdorff distance (MHD) based on the average distance value instead of the maximum value, in this sense they defined the directed distance of the MHD as

$$h(A,B) = \frac{1}{N_A} \sum_{a \in A} \min_{b \in B} \|a - b\| \tag{3.40}$$

Using these ingredients we can describe how Hausdorff distances can be extended to graph-based representations. To commence let us consider two graphs $G_1 = (V_1, E_1, k_1)$ and $G_2 = (V_2, E_2, k_2)$, where V_1,V_2 are the sets of nodes, E_1,E_2 the sets of edges and k_1,k_2 the matrices whose elements are the curvature defined in the previous section. We can now write the distances between two graphs as follows:

1) The classical Hausdorff distance (HD) is

$$h_{HD}(G_1, G_2) = \max_{i \in V_1} \max_{j \in V_1} \min_{I \in V_2} \min_{J \in V_2} \|k_2(I,J) - k_1(i,j)\| \tag{3.41}$$

2) The modified Hausdorff distance (MHD) is

$$h_{MHD}(G_1, G_2) = \frac{1}{|V_1|} \sum_{i \in V_1} (\frac{1}{|V_1|} \sum_{i \in V_1} \min_{I \in V_2} \min_{J \in V_2} \|k_2(I,J) - k_1(i,j)\|) \tag{3.42}$$

3.4.2 A probabilistic similarity measure (PSM)

One of the well documented problems with booth the Hausdorff and modified Hausdorff distances, is lack of robustness. In order to overcome this problem, Huet and Hancock (Heut & Hancock, 2002) have recently develop a probabilistic variant of the Hausdorff distance. This measures the similarity of the sets of

attributes rather than using defined set based distance measures. For the graphs G_1 and G_2, the set of all nodes connected to the node $I \in G_2$ by an edge is defined as $C_I^2 = \{J|(I,J) \in E_2\}$, and the corresponding set of nodes connected to the node $i \in G_1$ by an edge is $C_i^1 = \{j|(i,j) \in E_2\}$. For the match of the graph G_2 onto G_1 Huet and Hancock's similarity measure

$$S(G_1, G_2) = \frac{1}{|V_2| \times |V_1|} \sum_{i \in V_1} \max_{I \in V_2} \sum_{j \in C_i^1} \max_{J \in C_I^2} P((i,j) \rightarrow (I,J)|k^2_{(I,J)}, k^1_{(i,j)}) \tag{3.43}$$

In this formula the *a posteriori* probability $P((i,j) \rightarrow (I,J)|k^2_{(I,J)}, k^1_{(i,j)})$ represents the value for the match of the G_2 edge (I,J) onto the G_1 edge (i,j) provided by the corresponding pair of attribute structures $k^2_{(I,J)}$ and $k^1_{(i,j)}$.

The similarity measure commences by finding the maximum probability over the nodes in C_I^2 then averaging the edge-compatibilities over the nodes in C_i^1. Similarly, we consider the maximum probability over the nodes in the graph G_2 followed by averaging over the nodes in G_1. It is worth mentioning that unlike the Hausdorff distance, this similarity measure does not satisfy the metric axioms. Moreover, while the Hausdorff distance is saliency-based (i.e. it measures the maximum distance between two sets of observations) our measure here returns the maximum similarity. back to the formula where we still need to compute the probability $P((i,j) \rightarrow (I,J)|k^2_{I,J}, k^1_{i,j})$, for that purpose we will use a robust weighting function

$$P((i,j) \rightarrow (I,J)|k^2_{(I,J)}, k^1_{(i,j)}) = \frac{\Gamma_\sigma(\|k^2_{(I,J)}, k^1_{(i,j)}\|)}{\sum_{(I,J) \in E_2} \Gamma_\sigma(\|k^2_{(I,J)}, k^1_{(i,j)}\|)}) \tag{3.44}$$

where $\Gamma_\sigma(.)$ is a distance weighting function. There is several alternative robust weighting functions. Here we work with a Gaussian of the form $\Gamma_\sigma(\rho) = \exp(-\frac{\rho^2}{2\sigma^2})$.

3.4.3 Multidimensional Scaling

The multidimensional scaling (MDS) is a technique to provide a visual representation of the pattern of proximities (i.e., similarities or distances) among a set of objects. The input to MDS is a square, symmetric matrix indicating dissimilarities between pairs of objects. Here the objects are represented as points in a low dimensional space, such that the distances between the points match the observed dissimilarities as closely as possible. As a starting point, let H be the distance matrix with row r and column c entry H_{rc}. The first step of MDS is to calculate a matrix T whose element with row r and column c is given by $T_{rc} = -\frac{1}{2}[H_{rc}^2 - \widehat{H}_{r.}^2 - \widehat{H}_{.c}^2 + \widehat{H}_{..}^2]$ where $\widehat{H}_{r.} = \frac{1}{N}\sum_{c=1}^{N} H_{rc}$ is the average value over the rth row in the distance matrix, $H_{.c}$ is the similarly defined average value over the cth column and $\widehat{H}_{..} = \frac{1}{N^2}\sum_{r=1}^{N}\sum_{c=1}^{N} H_{rc}$ is the average value over all rows and columns of the distance matrix. Then, we subject the matrix T to an eigenvector analysis to obtain a matrix of embedding coordinates X. If the rank of T is k; $k \leq N$, then we will have k non-zero eigenvalues. We arrange these k non-zero eigenvalues in descending order, i.e., $l_1 \geq l_2 \geq ... \geq l_k \geq 0$. The corresponding ordered eigenvectors are denoted by u_i where l_i is the ith eigenvalue. The embedding coordinate system for the graphs is $X = [\sqrt{l_1}u_1, \sqrt{l_2}u_2, ..., \sqrt{l_k}u_k]$ for the graph indexed i, the embedded vector of the coordinates is $x_i = (X_{i,1}, X_{i,2}, ..., X_{i,k})^T$.

3.4.4 Discerption of the Experimental Databases

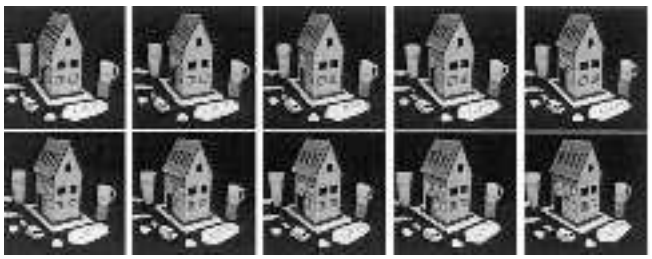
Through our experiments we are going to use three different sets of data. This section is devoting for a quick description of these databases, the houses database 3.4.4.1, The COIL database 3.4.4.2 and the Toys database 3.4.4.3.

3.4.4.1 The York model house dataset

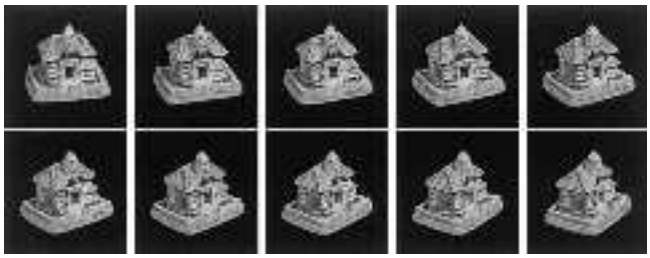
The first dataset is the York model house database, which contains different graphs extracted from images of toy houses in the standard CMU, MOVI and chalet house image sequences (Luo et al., 2003). These data sets contain different views of model houses from equally spaced viewing directions. From the house images, corner features are extracted using the corner detector reported in (Luo, Cross & Hancock, 1999), and Delaunay graphs representing the arrangement of feature points are constructed. This data consists of ten graphs for each of the three houses. Each node in a Delaunay graph belongs to a first order cycle, and as a result the graph is a triangulation. The images of the houses and their associated Delaunay triangulations are shown in Figure 3.3. In figures 3.4, 3.5 and 3.6 we show the node, edge and face frequencies for the houses database. Tables 3.1, 3.2 and 3.3 contain the number of nodes, edges and triangulated faces respectively.

In the database, the different graphs have different number of nodes and one can notice that the INRIA MOVI sequence contains many more feature points than the other two sequences and there is a little texture in its image sequences comparing to the other images which might led the corner detection used to fail extracting the graph features.

(a) CMU/VASC model house sequence.



(b) INRIA MOVI model house sequence.



(c) The Swiss chalet model house sequence.

Figure 3.3: Sample images from the houses image sequences with the extracted graphs.

	CMU	MOVI	chalet
v1	30	140	40
v2	32	134	57
v3	32	130	92
v4	30	136	78
v5	30	137	90
v6	32	131	64
v7	30	139	113
v8	30	141	100
v9	30	133	67
v10	31	136	59

Table 3.1: The number of Nodes of the graphs from the houses database.

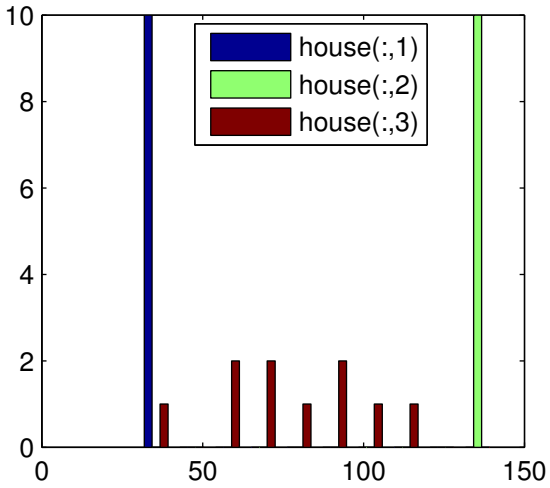


Figure 3.4: Histogram of the number of Nodes of the graphs from the houses database.

	CMU	MOVI	chalet
v1	158	808	216
v2	168	780	312
v3	168	756	516
v4	156	792	438
v5	156	796	512
v6	170	760	358
v7	156	808	638
v8	156	812	562
v9	156	772	370
v10	164	786	324

Table 3.2: The number of Edges of the graphs from the houses database.

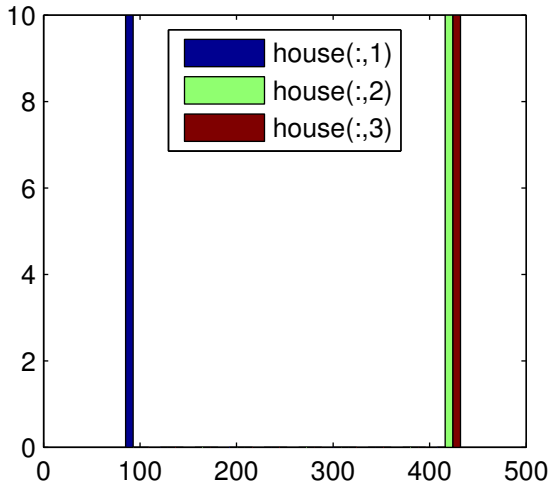


Figure 3.5: Histogram of the number of Edges of the graphs from the houses database.

	CMU	MOVI	chalet
v1	50	265	69
v2	55	257	100
v3	53	249	172
v4	49	261	142
v5	49	262	167
v6	54	250	116
v7	49	266	207
v8	49	266	182
v9	49	254	119
v10	52	258	104

Table 3.3: The number of Faces of the graphs from the houses database.

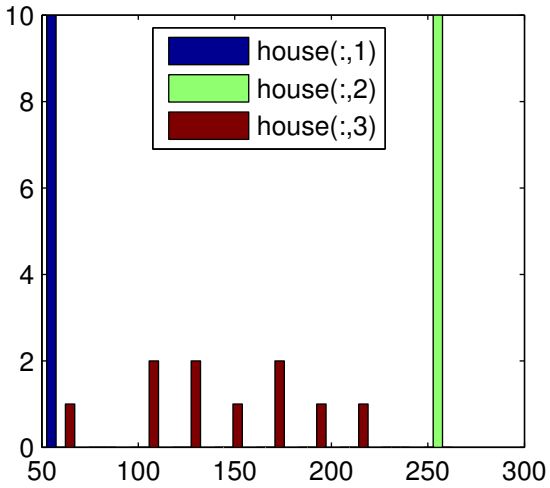


Figure 3.6: Histogram of the number of Triangular Faces of the graphs from the houses database

3.4.4.2 The COIL dataset

The second database we are going to use for our experiments is the Columbia Object Image Library (COIL-20) database (Figure 3.7).

Figure 3.7: The Columbia Object Image Library (COIL-20).

The objects have a wide variety of complex geometric and reflectance characteristics. COIL-20 is a database of 1,440 gray-scale images of 20 objects. The objects were placed on a motorized turntable against a black background. Each object was placed in a stable configuration at approximately the centre of the table. Then the turntable was rotated through 360 degrees to vary object pose with respect to a fixed camera. Images of the objects were taken at pose intervals of 5 degrees; this corresponds to 72 images per object. The images were also normalized such that the larger of the two object dimensions (height and width) fits the image size off 128 x 128 pixels. When resizing, aspect ratio was preserved. In addition to size normalization, every image was histogram stretched, i.e. the intensity of the brightest pixel was made 255 and intensities of the other pixels were scaled accordingly. Consequently, the apparent scale of the object may change between different views of the object image especially for the objects which are not symmetric with respect to the turntable axis. The (COIL-20) dataset is available online via ftp in addition to the (COIL-100) dataset of colour images of 100 objects (Figure 3.8), that is 7,200 poses in total (COIL is available

at http://www.cs.columbia.edu/CAVE/databases/).

Figure 3.8: The Columbia Object Image Library (COIL-100).

The frequencies of the nodes, edges and triangular faces of each graph in the COIL database is shown in the Figures 3.9, 3.10 and 3.11, respectively. In A.3, a full tables containing the number of nodes, edges and triangulated faces respectively.

From the data, we know that the COIL dataset consists of a large number of objects with varying pose, texture, shape and size. This might lead to a difficulty in recognizing the objects, hence many recognition methods use 36 (10 degrees apart) of them for training and the remaining images for testing. Nevertheless, we will use a smaller set of 18 views (20 degrees apart) per object to obtain better variations.

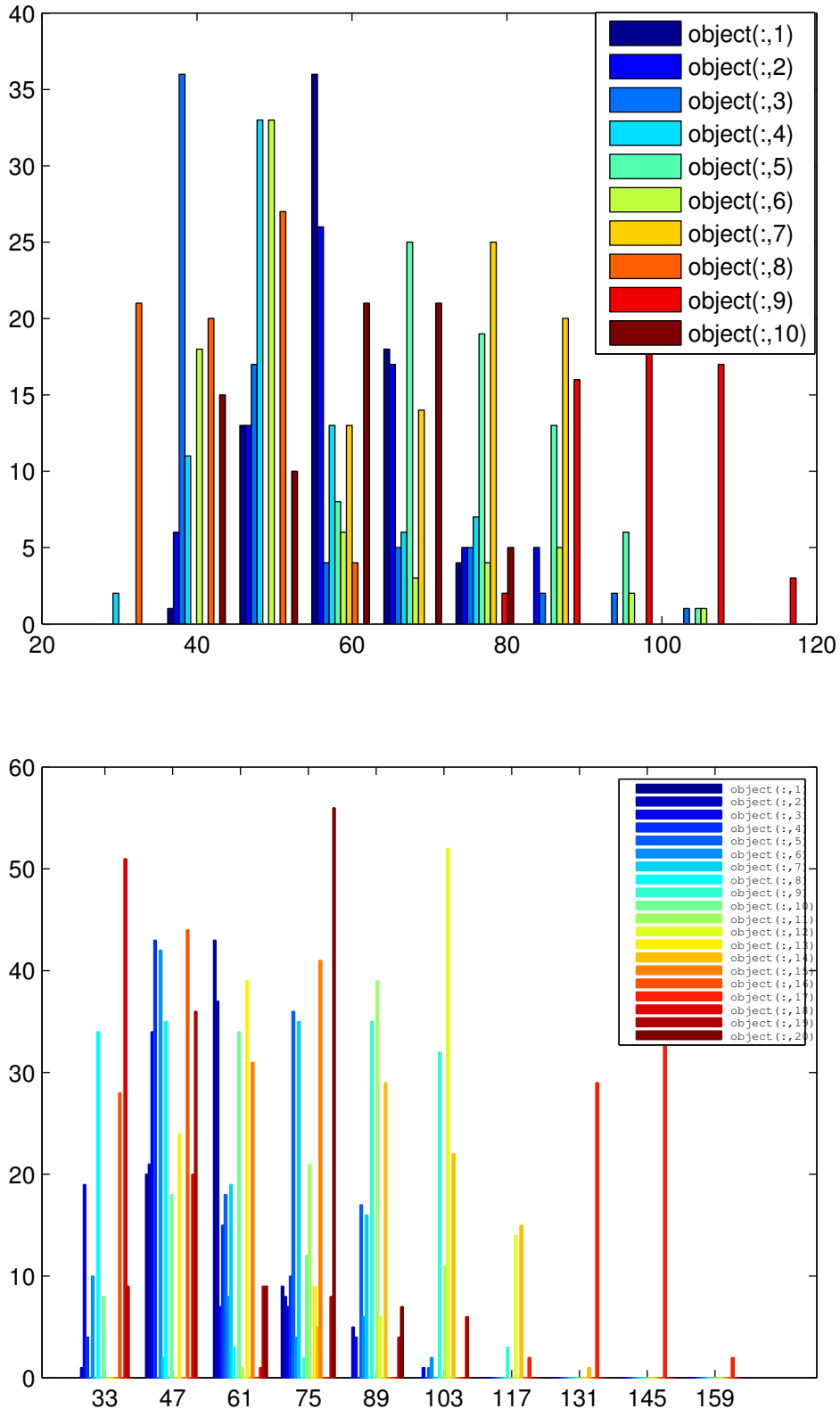


Figure 3.9: Histogram of The number of Nodes for 10 objects (top) and 20 objects (bottom) of the COIL data.

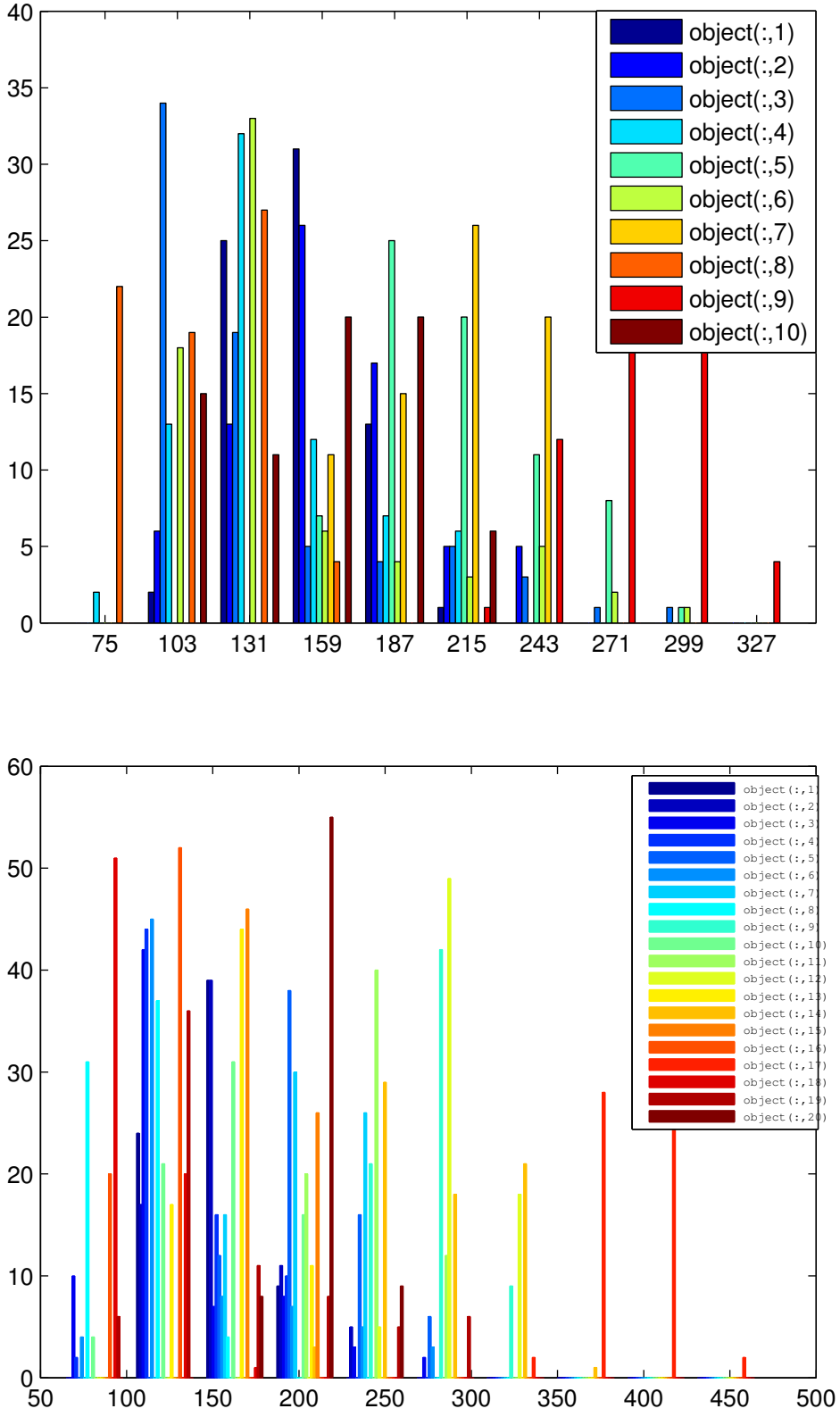


Figure 3.10: Histogram of The number of Edges for 10 objects (top) and 20 objects (bottom) of the COIL data.

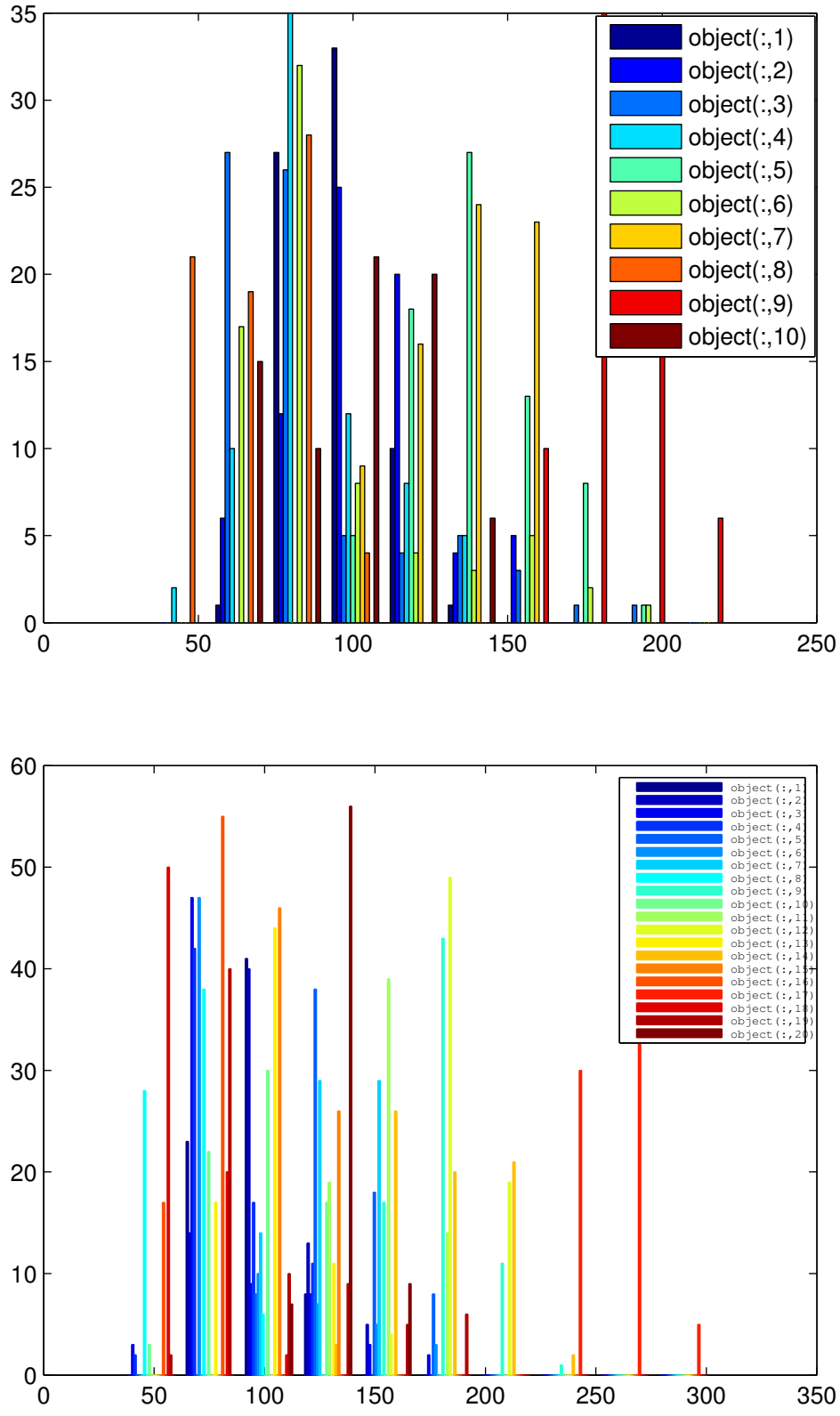


Figure 3.11: Histogram of The number of Triangular Faces for 10 objects (top) and 20 objects (bottom) of the COIL data.

3.4.4.3 The Toy dataset

The third dataset to be used through this thesis is The Toy Database (Han, Wilson & Hancock, 2010). This dataset consists of images of 4 objects with 20 different views of each object. Figure 3.12 shows example images of the 4 objects. The feature keypoints in the images are extracted using the SIFT detector (Lowe, 2004) and the sample graphs are constructed using Delaunay triangulation of the detected points.

Figure 3.12: Example images of the 4 objects of The Toy Database.

The frequencies of the nodes and edges of each graph in the Toy database is shown in the Figures 3.13 and 3.14, respectively. As well, the number of of nodes and edges of each graph are contained in the Tables 3.4 and 3.5, respectively.

In general, the database contains large and noisy graphs with different numbers of nodes.

	T1	T2	T3	T4
v1	123	111	126	99
v2	71	91	87	74
v3	69	89	102	83
v4	70	100	108	89
v5	70	104	98	79
v6	68	100	101	77
v7	77	96	100	70
v8	82	99	101	74
v9	82	93	104	71
v10	92	87	106	80
v11	97	95	94	87
v12	107	88	98	85
v13	106	88	106	89
v14	102	91	106	88
v15	104	98	102	86
v16	111	91	100	82
v17	106	91	85	83
v18	108	92	102	83
v19	115	92	91	78
v20	162	106	132	103

Table 3.4: The number of Nodes of the graphs from the Toy database.

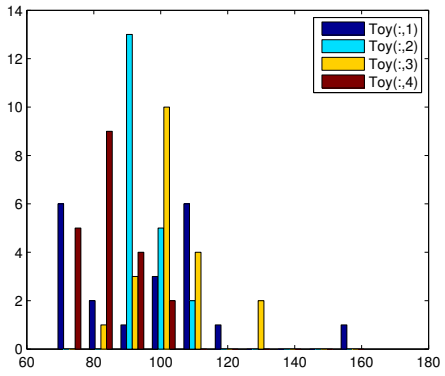


Figure 3.13: Histogram of the number of Nodes of the graphs from the Toy database

	T1	T2	T3	T4
v1	355	318	348	287
v2	361	318	350	292
v3	362	320	353	290
v4	360	317	347	289
v5	362	320	347	289
v6	363	322	349	287
v7	354	317	350	278
v8	357	319	350	267
v9	360	322	353	258
v10	361	302	348	256
v11	362	312	340	260
v12	360	261	319	243
v13	359	259	343	254
v14	367	265	344	250
v15	363	290	337	243
v16	357	257	317	240
v17	360	259	273	238
v18	360	261	324	238
v19	362	262	282	224
v20	473	305	381	298

Table 3.5: The number of Edges of the graphs from the Toy database.

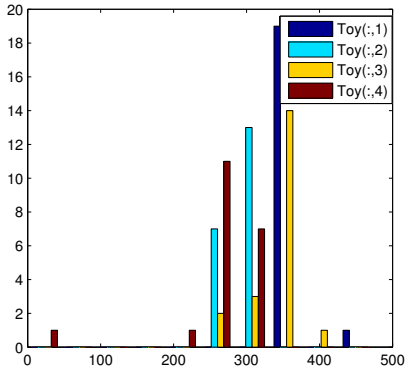


Figure 3.14: Histogram of the number of Edges of the graphs from the Toy database

3.4.5 Experimenting with Real-world data

In this section we experiment with the curvature-based attributes extracted using the heat-kernel embedding, then we explore whether these attributes can be used for the purposes of graph-matching. To do so we follow the next steps:

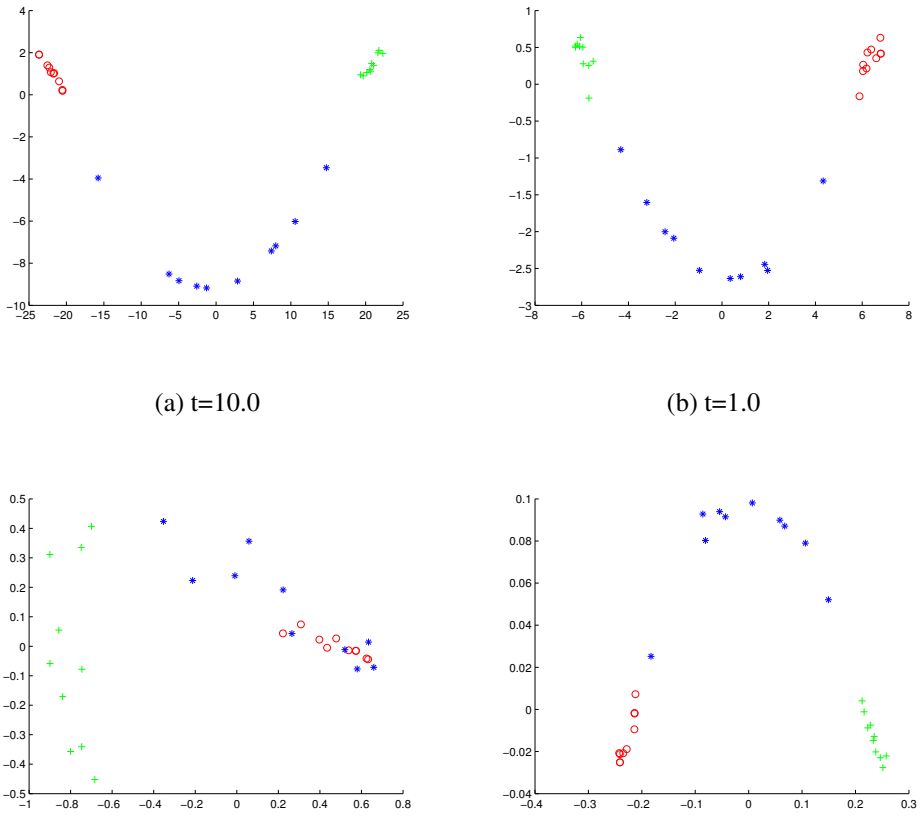
- First, with the adjacency matrices in hand we commence by constructing the Normalised Laplacian matrix for each graph in the database.
- Then we use the Heat kernel embedding defined in Section 3.2.4 to embed the nodes of the graphs into points residing on a manifold in a Euclidean space.
- The Euclidean distance between pairs of points in the Euclidean space is obtained from the heat kernel embedding at the values of $t = 10.0, 1.0, 0.1$ and 0.01 using formula 3.14.
- At this point, we construct the matrix representing each graph based on the geometric attributes deduced earlier in this chapter.

We proceed with two representations for the graphs. The first is the sectional curvature associated with the edges, outlined in Section 3.3.1. The second is the Gaussian curvature on the triangles of the Delaunay triangulations extracted from the graphs, as outlined in Section 3.3.2.

Both the sectional and gaussian curvature will be used as graph features for the purposes of gauging the similarity of graphs using the Hausdorff distance and a robust modified variant of the Hausdorff distance (given in Section 3.4.1) as well as the probabilistic similarity measure (given in Section 3.4.2). Finally we subject the distance matrices to the Multidimensional Scaling (MDS) procedure (given in Section 3.4.3) to embed the graphs into a low dimensional space where each graph is represented as a single point in a 2D space.

We commence by introducing the results obtained when experimenting with the York model house database. First, we show in Figures 3.15 and 3.16 the results when using the Hausdorff distance (HD) to measure the (dis)similarity

between pairs of graphs represented by the Sectional and Gaussian curvatures respectively. The subfigures are ordered from left to right, top to bottom using the heat kernel embedding with the values $t = 10.0, 1.0, 0.1$ and 0.01 respectively. With the same order, Figures 3.17 and 3.18, give the results obtained when using the Modified Hausdorff distance (MHD) and Figures 3.19, 3.20 stands for the results obtained when using the probabilistic similarity measure (PSM). In all figures each graph of the CMU model house sequence is represented as a red circle and each graph of the MOVI model house sequence is represented as blue star while each graph of the Swiss chalet model house sequence is represented as a green cross.



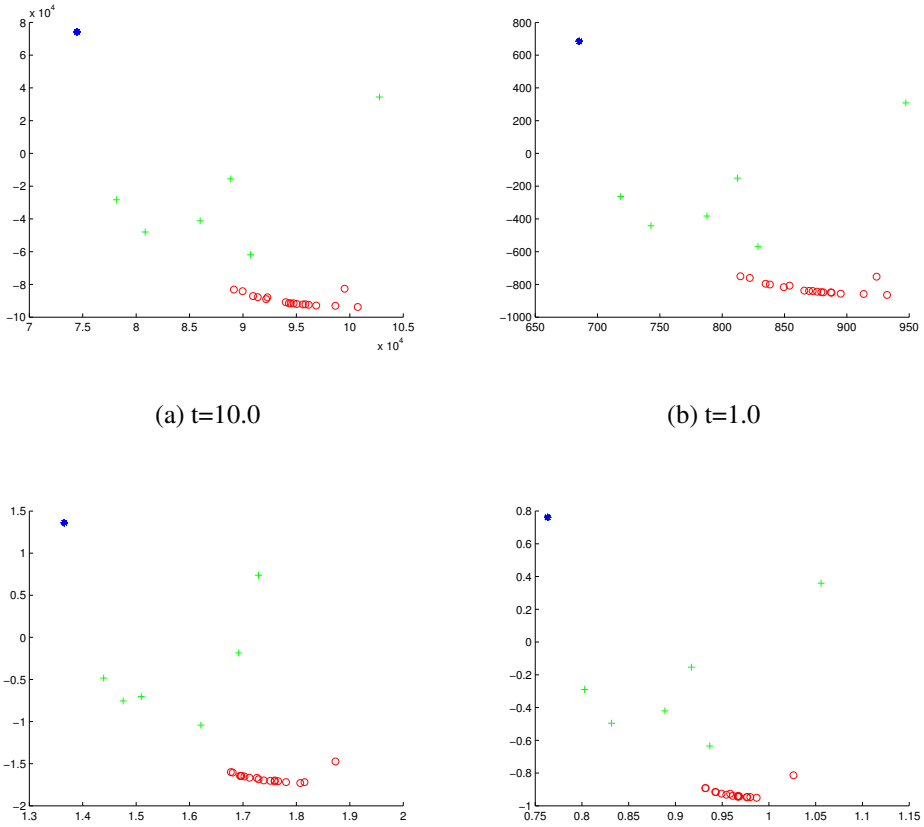
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.15: MDS embedding obtained using HD for house data represented by the sectional curvatures residing on the edges.



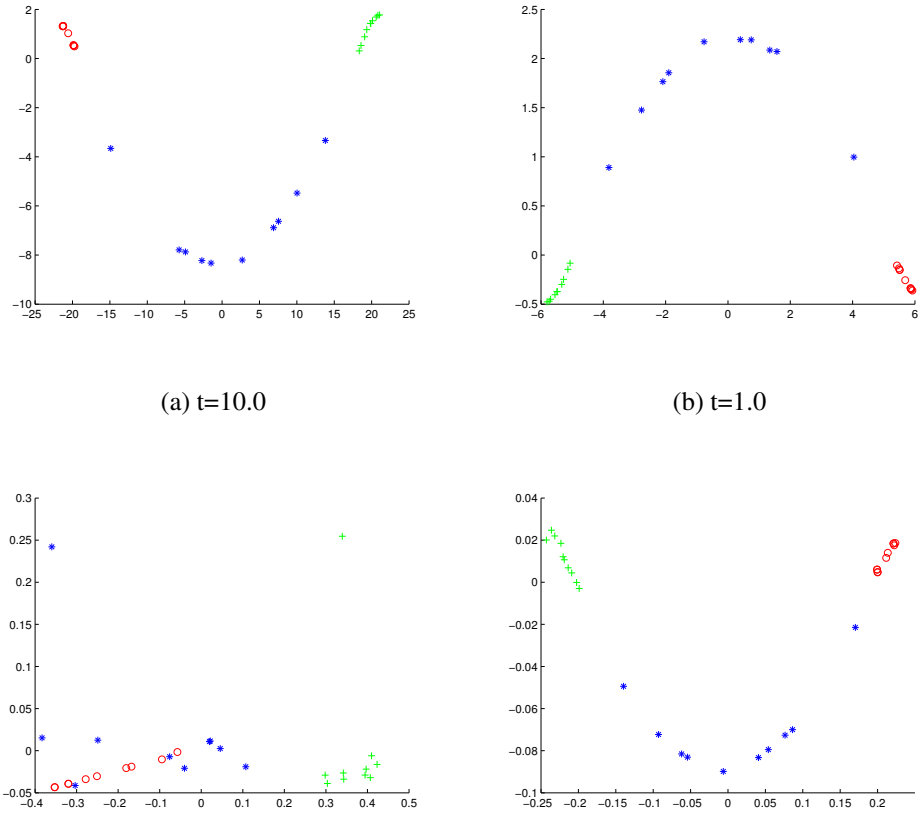
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.16: MDS embedding obtained using HD for the houses data represented by the Gaussian curvature associated with the geodesic triangles.



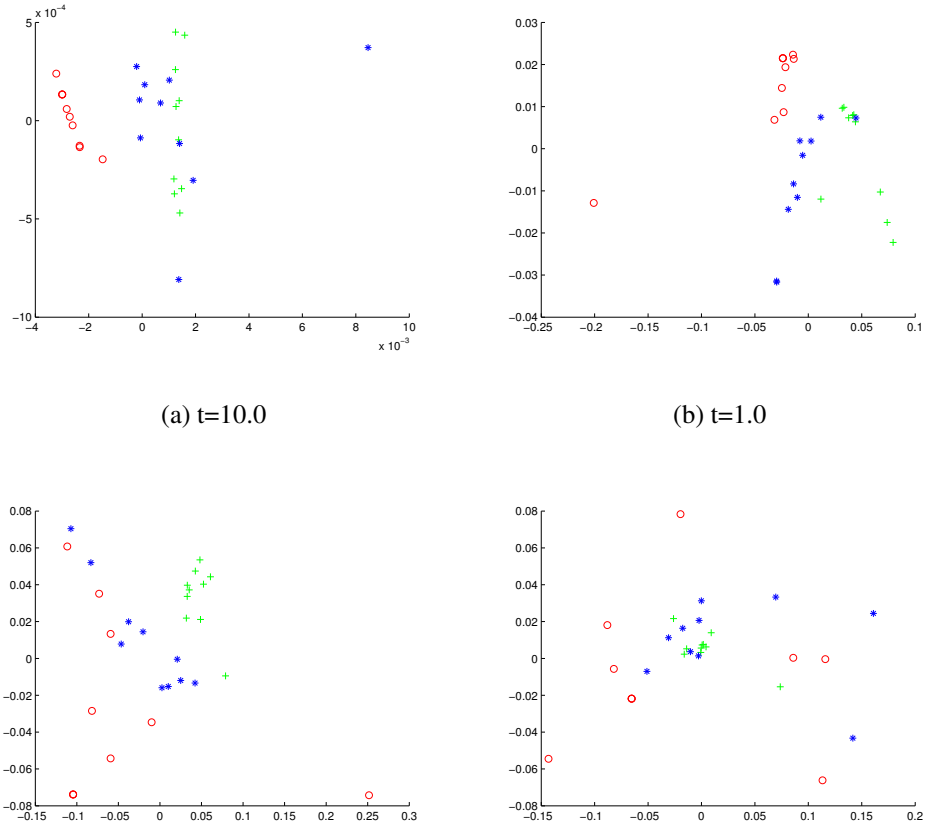
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.17: MDS embedding obtained using MHD for house data represented by the sectional curvatures residing on the edges.



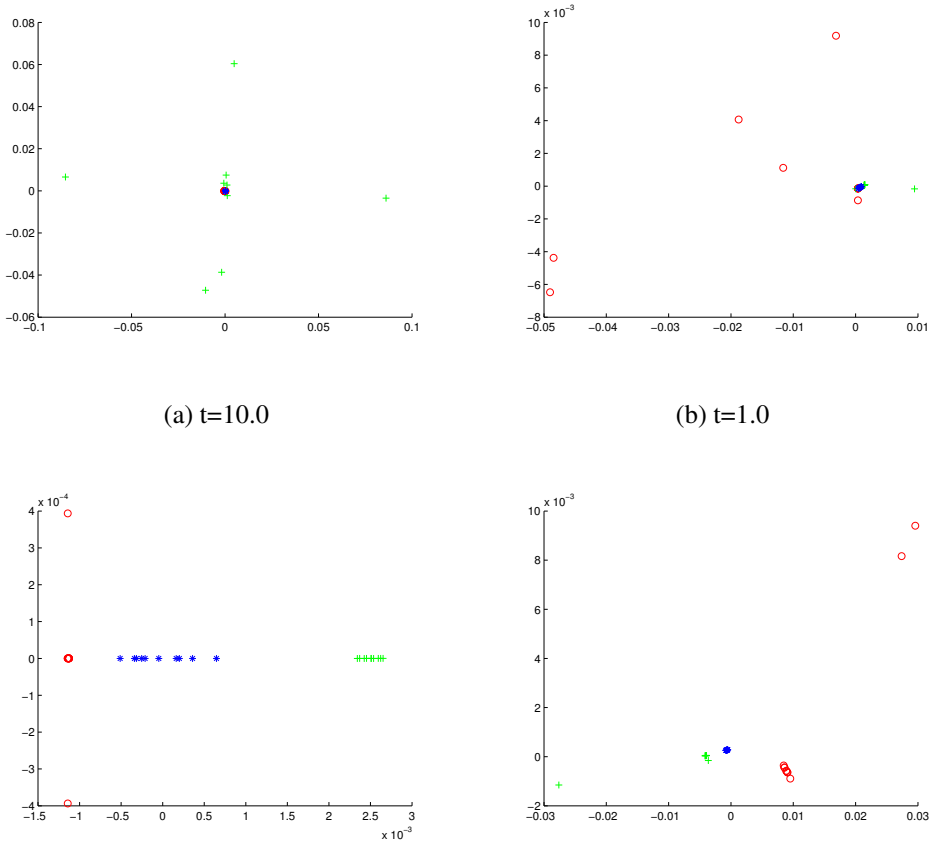
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.18: MDS embedding obtained using MHD for the houses data represented by the Gaussian curvature associated with the geodesic triangles.



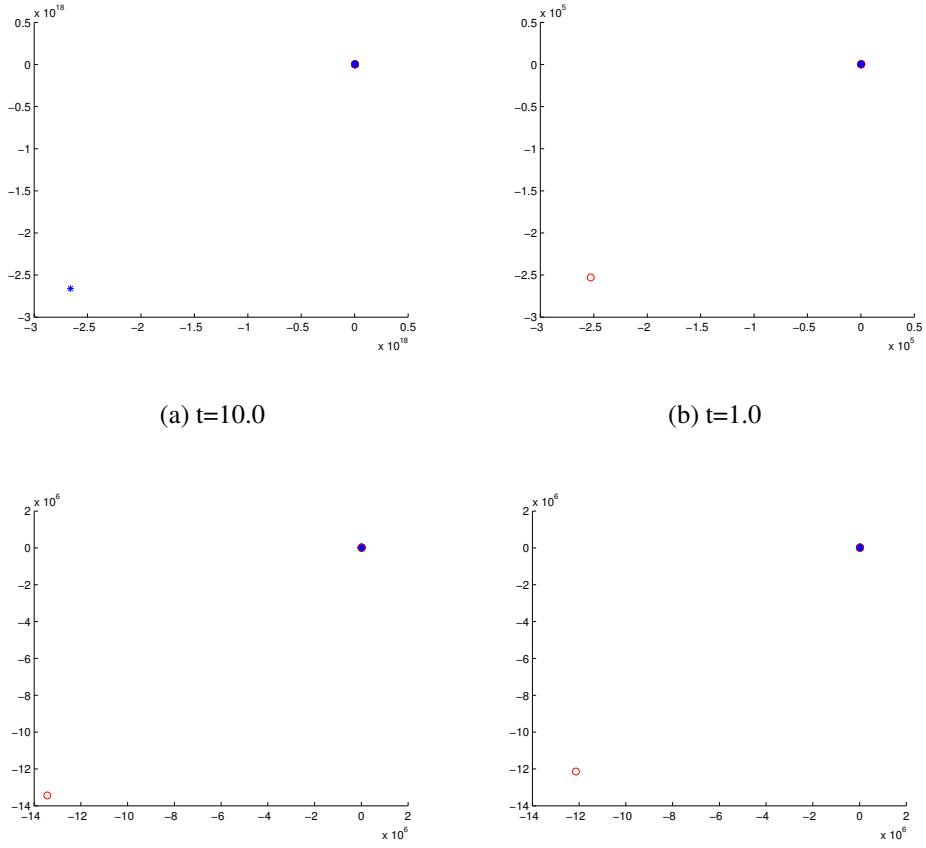
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.19: MDS embedding obtained using the probabilistic similarity measure for the houses data set represented by the sectional curvature residing on the edges.



(a) t=10.0 (b) t=1.0

(c) t=0.1 (d) t=0.01

Figure 3.20: MDS embedding obtained using the probabilistic similarity measure for the houses data set represented by the Gaussian curvature associated with the geodesic triangles.

To investigate the data in more detail Table 5.1 shows the rand index for the data as a function of t. This index is computed as follows:

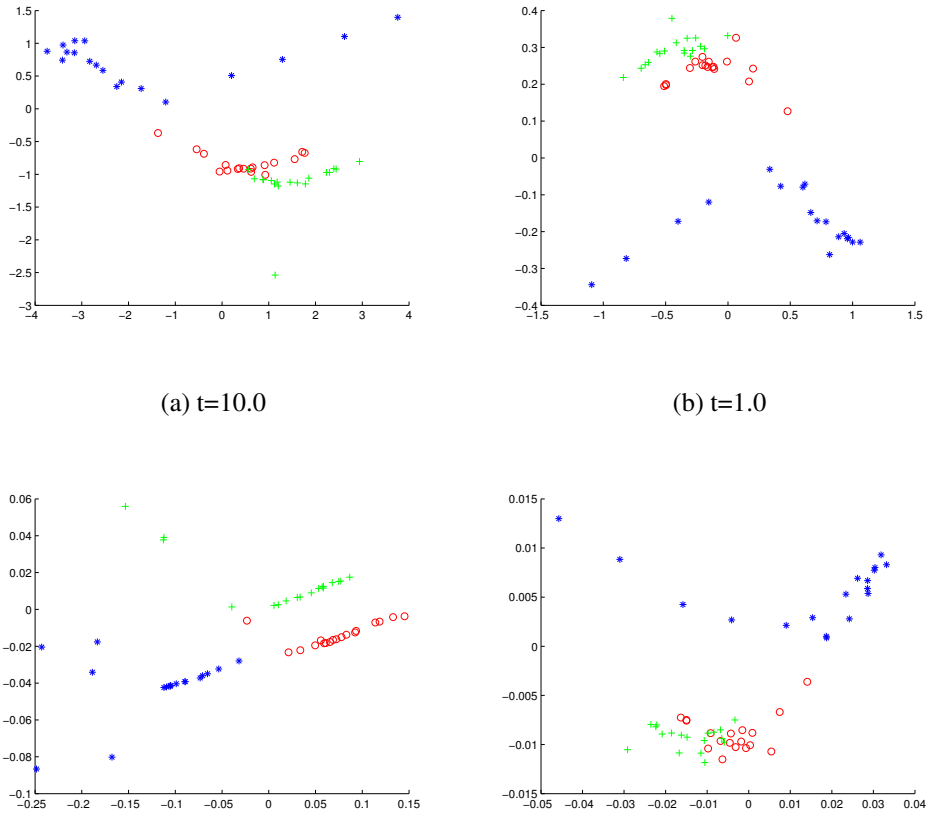
- We commence by computing the mean for each cluster.
- Then we compute the distance from each point to each mean.
- If the distance from the correct mean is smaller than those to remaining means, then the classification is correct, if not then the classification is incorrect.
- The rand index is

  R= (♯ incorrect) / (♯ incorrect + ♯ correct).

		t=10	t=1.0	t=0.1	t=0.01
HD	Sectional curvature	0.1000	0.1667	0.4333	0.0333
HD	Gaussian curvature	0.5000	0.1333	0.1000	0.5000
MHD	Sectional curvature	0.1333	0.2333	0.1333	0.0333
MHD	Gaussian curvature	0.1667	0.0333	0.1333	0.4000
PSM	Sectional curvature	0.0000	0.0333	0.2667	0.3667
PSM	Gaussian curvature	0.3000	0.3000	0.3000	0.3000

Table 3.6: A rand index vs. t for York model house database

Now, we give the results obtained when experimenting with three objects from the COIL database. Figures 3.21 and 3.23 show the results when using the Hausdorff distance (HD) to measure the (dis)similarity between pairs of graphs represented by the Sectional and Gaussian curvatures respectively. Once again the distance matrices are subjected to the Multidimensional Scaling (MDS) procedure (given in Section 3.4.3) to embed the graphs into a low dimensional space representing each graph as a single point in a 2D space; where each graph of the sequence of the first object is represented as a red circle and each graph of the sequence of the second object is represented as blue star while each graph of the sequence of the third object is represented as a green cross. The subfigures are ordered from left to right, top to bottom using the heat kernel embedding with the values $t = 10.0, 1.0, 0.1$ and 0.01 respectively. With the same order, Figures 3.22 and 3.24 give the results obtained when using the Modified Hausdorff distance (MHD). Followed by Table 3.7 giving the Rand Index for the data as a function of t.



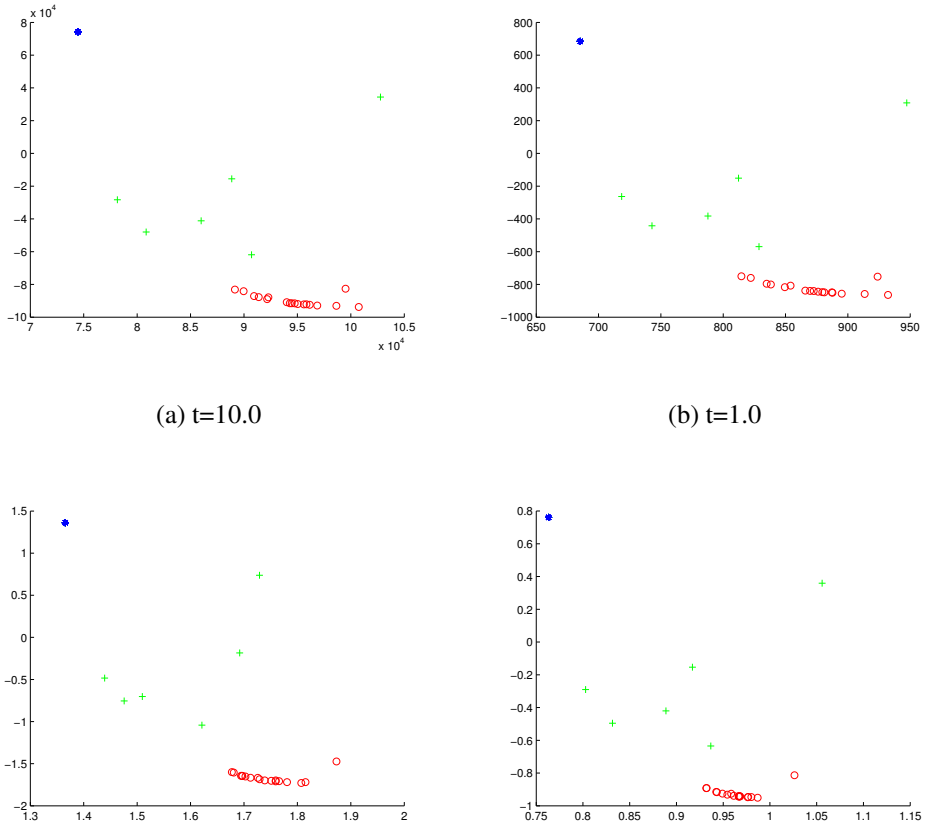
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.21: MDS embedding obtained using HD for COIL data represented by the sectional curvatures residing on the edges.



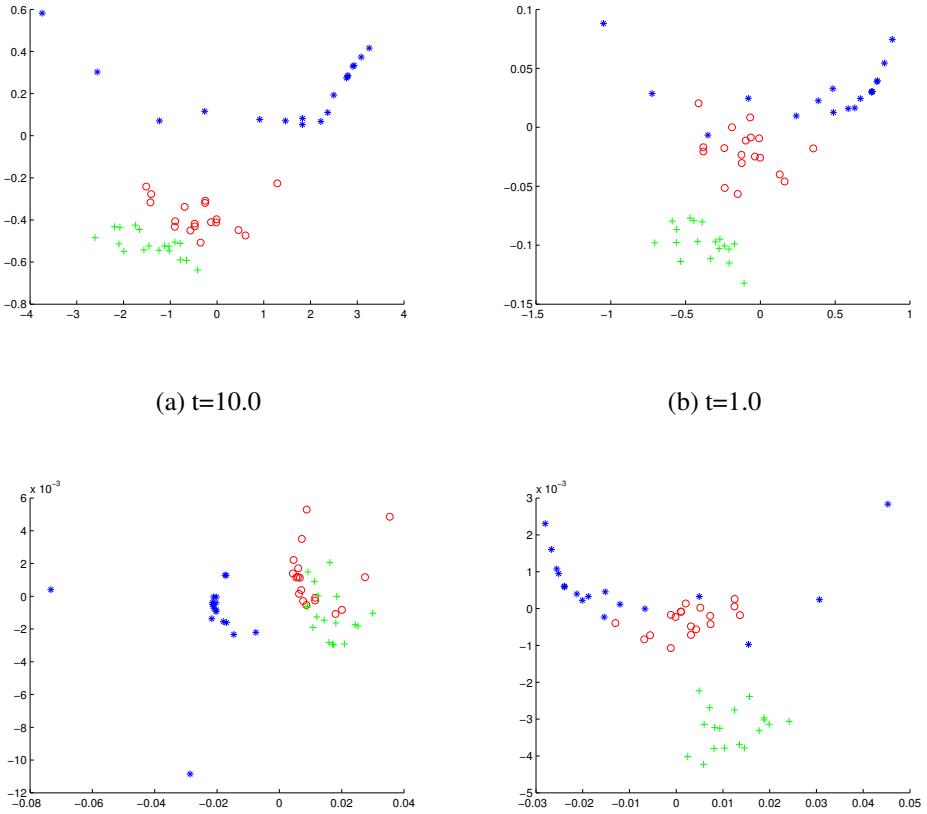
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.22: MDS embedding obtained using HD for COIL data represented by the Gaussian curvatures associated with each node.



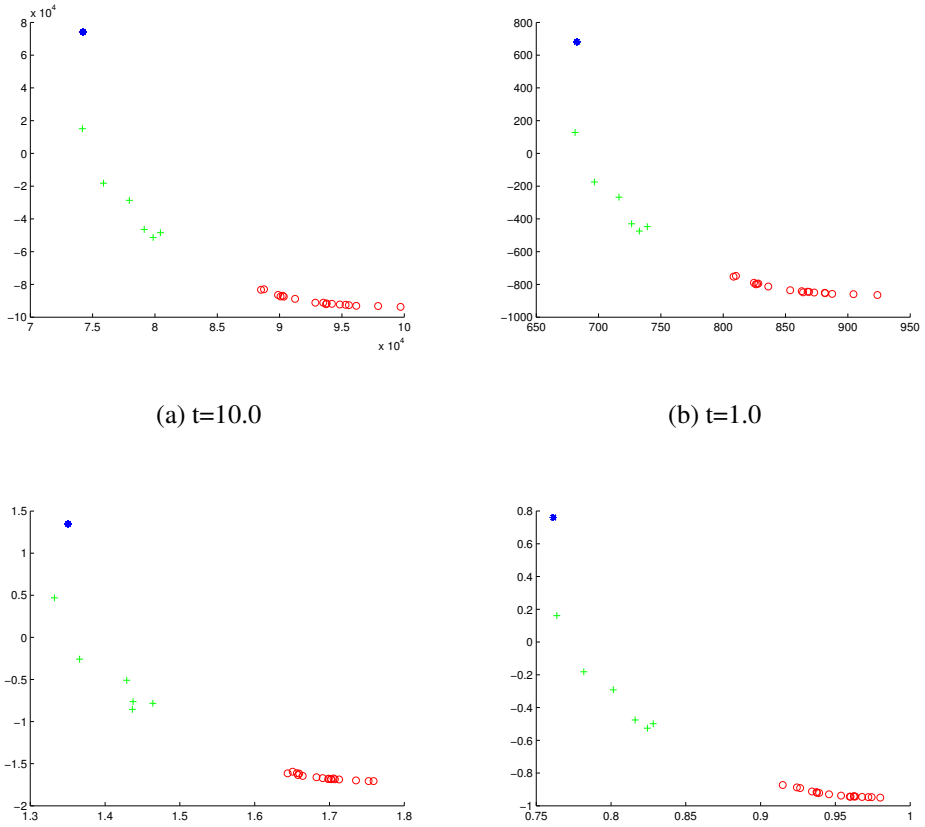
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.23: MDS embedding obtained using MHD for COIL data represented by the sectional curvatures residing on the edges.



(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.24: MDS embedding obtained using MHD for COIL data represented by the Gaussian curvatures associated with each node.

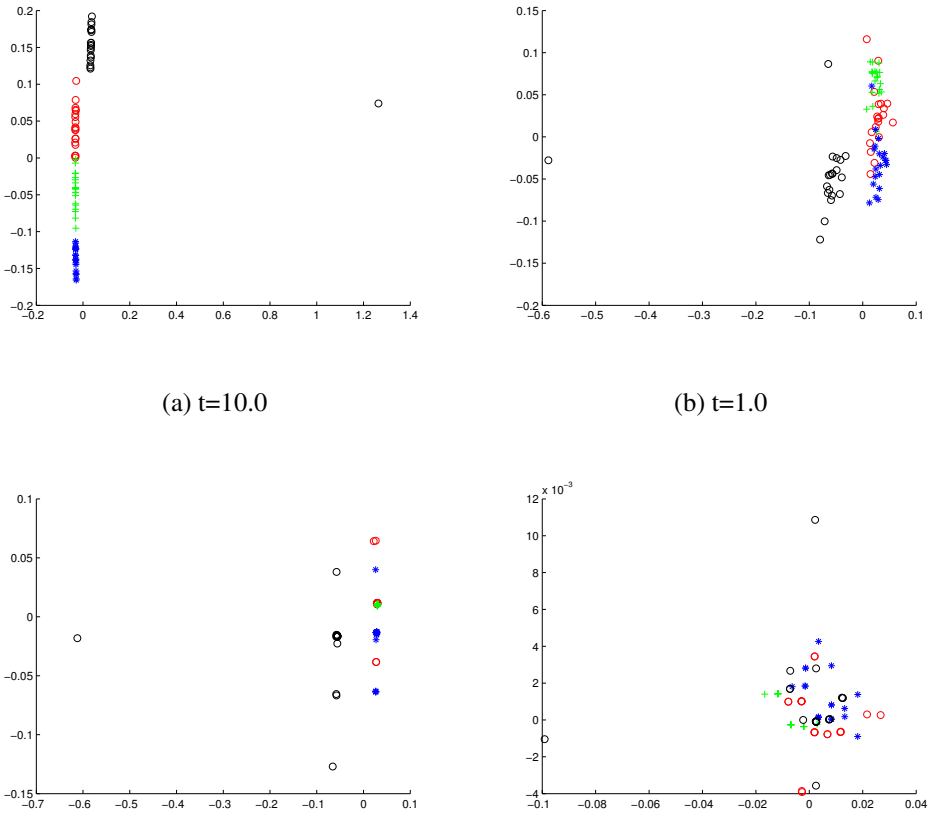
		t=10	t=1.0	t=0.1	t=0.01
HD	Sectional curvature	0.1667	0.2037	0.2407	0.2037
HD	Gaussian curvature	0.2222	0.0000	0.0000	0.2222
MHD	Sectional curvature	0.1852	0.1852	0.1667	0.2222
MHD	Gaussian curvature	0.2222	0.0926	0.0000	0.2222

Table 3.7: A rand index vs. t for COIL database

For more experiments we show the results obtained when using three objects from the TOY database. With the same technique used with the other two datasets each graph is represented as a single point in a 2D space. where each graph of the sequence of the first object is represented as a red circle and each graph of the sequence of the second object is represented as blue star while each graph of the sequence of the third object is represented as a green cross. Figure 3.25 show the results when using the Hausdorff distance (HD) to measure the (dis)similarity between pairs of graphs represented by the Sectional Curvature. The subfigures are ordered from left to right, top to bottom using the heat kernel embedding with the values $t = 10.0, 1.0, 0.1$ and 0.01 respectively. With the same order, Figures 3.26 gives the results obtained when using the Modified Hausdorff distance (MHD). Table 3.8 shows the Rand Index for the data as a function of t.

		t=10	t=1.0	t=0.1	t=0.01
HD	Sectional curvature	0.0333	0.2167	0.0833	0.5000
MHD	Sectional curvature	0.0003	0.1500	0.2833	0.4500

Table 3.8: A rand index vs. t for TOY database



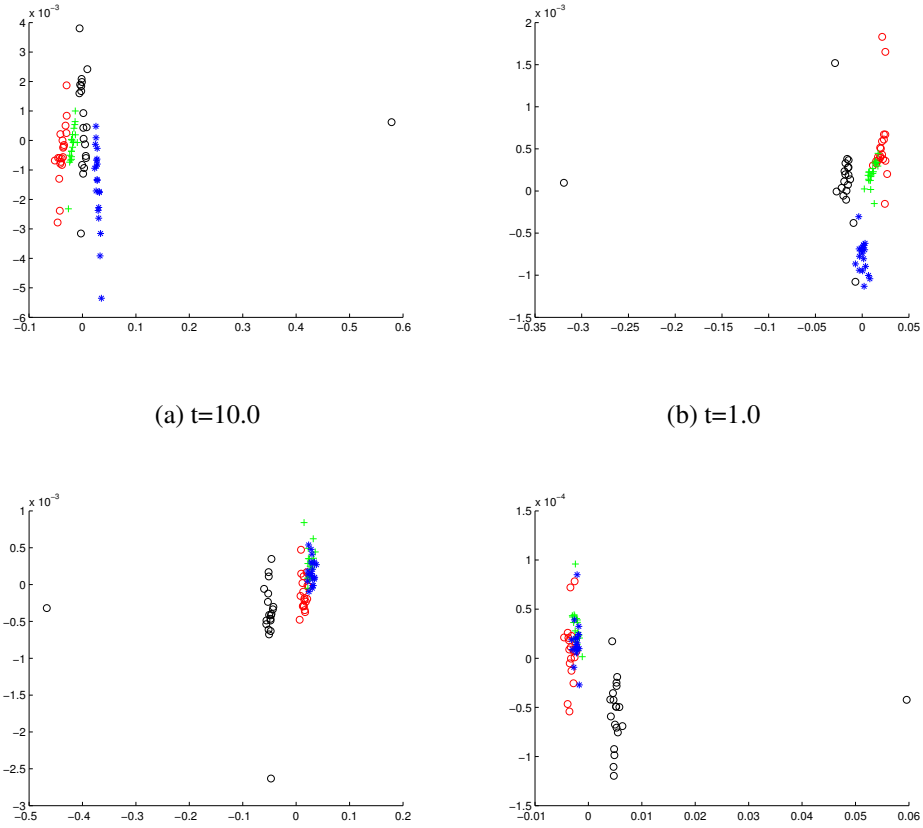
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.25: MDS embedding obtained using HD for TOY data represented by the sectional curvatures residing on the edges.



(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 3.26: MDS embedding obtained using MHD for TOY data represented by the sectional curvatures residing on the edges.

There are several observations which can be derived from the previous figures. For instance, when experimenting with the Sectional curvature for the York model house's dataset, the MDS results produced by HD and MHD distance measures are essentially one-dimensional, i.e. the data are scaled around a tight curve. In the case of the probabilistic similarity measures, the separation is clearest when $t = 0.1$. Here the data are clustered along a Straight line. Moreover, it was shown that the clusters of the MOVI model house's sequence is less compact than the other two clusters (CMU and Swiss chalet model houses' sequences), which might be happened due to the more feature points and the little texture contained in the images' sequences of the MOVI house rather than the other two sequences as one can see in Tables 3.1, 3.2 and 3.3, which show the number of nodes, edges and triangulated faces respectively. On the contrary, when experimenting with the Gaussian curvature the MOVI sequence is embedded around one point, however the clusters of the CMU and Swiss chalet model houses' sequences are well distinguished specially for value of $t = 10.0$ and $t = 1.0$. Furthermore, when using the probabilistic similarity measure the clusters are more compact that is each cluster is represented as a single point, where the idea here is to compute the similarity between the graphs rather than the dis-similarity as in the case of the HD and MHD measures. As for the COIL dataset, the experiments show that the situation is totally the opposite where the data are scaled around a curve when experimenting with the Gaussian curvature while there is no specific pattern appears when using the Sectional curvature; however the clusters are clear especially at $t = 10$, and where the same circumstances hold for the TOY dataset. It is worth mentioning that when we use values of t which is greater than 10, the obtained pattern is very much like the pattern obtained when $t = 10$; this resemblance can be understood when we know that the exponential function with a negative power (which is the base of the heat kernel) tends to zero as the value of t grows.

We end up this chapter, by investigating how the Gaussian curvatures of the geodesic triangles are distributed over the Delaunay graph of each view of the houses from the York model houses dataset. The curvature of each triangle in a

colour scale ranging from negative to positive values , is given in Figures, 3.27 through 3.35. Figures 3.27, 3.28 and 3.29, show the distribution of sample embeddings computed using the heat kernel embedding when $t = 1.0$. While, Figures 3.30, 3.31 and 3.32) stands for the same computation when $t = 0.1$. Finally, the results when $t = 0.01$ are shown in Figures 3.33, 3.34 and 3.35.

From the sequence it is clear that the Gaussian curvature distribution over the different views of each house is stable; moving smoothly from positive (elliptical) to negative (hyperbolic) regions. Moreover, when $t = 1.0$ all the Gaussian curvatures were positive which give an explanation for the MDS results obtained previously; that is in our embeddings we forced the data to be positive by projecting it in a positive space. This suggests that the arrangement of triangles and their Gaussian curvatures could be used as the basic of a matching algorithm particularly when using values of t equal to or greater than 1.

3.5 Conclusion

In this chapter, we aimed to investigate whether we can use the heat kernel to provide a geometric characterization of graphs to be used for the purposes of graph matching and clustering. This is a problem which can be addressed directly by using the spectral geometry of the combinatorial Laplacian. Performing Young-Householder decomposition to the heat-kernel maps the nodes of the graph to points in the manifold providing a matrix of the embedding co-ordinates. Assuming that the manifold on which the nodes of the graph reside is locally Euclidean, the heat kernel is approximated by a Gaussian function of the geodesic distance between nodes. Then the Euclidean distances between the nodes of the graph under study is estimated by equating the spectral and Gaussian forms of the heat kernel and the geodesic distance (that is the shortest distance on the manifold) is given by the floor of the path-length distribution, which can be computed from the Laplacian spectrum. With the embeddings to hand, we developed a graph characterization based on differential geometry. To do so we computed the sectional curvatures associated with the edges of the graph, making use of the fact that the sectional curvature can be determined by the difference between the geodesic and Euclidean distances between pairs of nodes. Taking this analysis one step further, we used the Gauss-Bonnet theorem to compute the Gaussian curvatures associated with triangular faces of the graph.

Characterizing the graphs using sets of curvatures, defined either on the edges or the faces, we explored whether these characterizations can be used for the purpose of graph matching and clustering. To this end, we compute the similarities of the sets using robust variants of the Hausdorff distance which allows us to compute the similarity of different graphs without knowing the correspondences between graph edges or faces.

In the section of experiments, results are provided for both Sectional and Gaussian curvatures characterizations of the graphs. The two characterizations were used for gauging the graph similarity. The databases used for that purpose were the York model house , some selected items from the COIL-20 and the

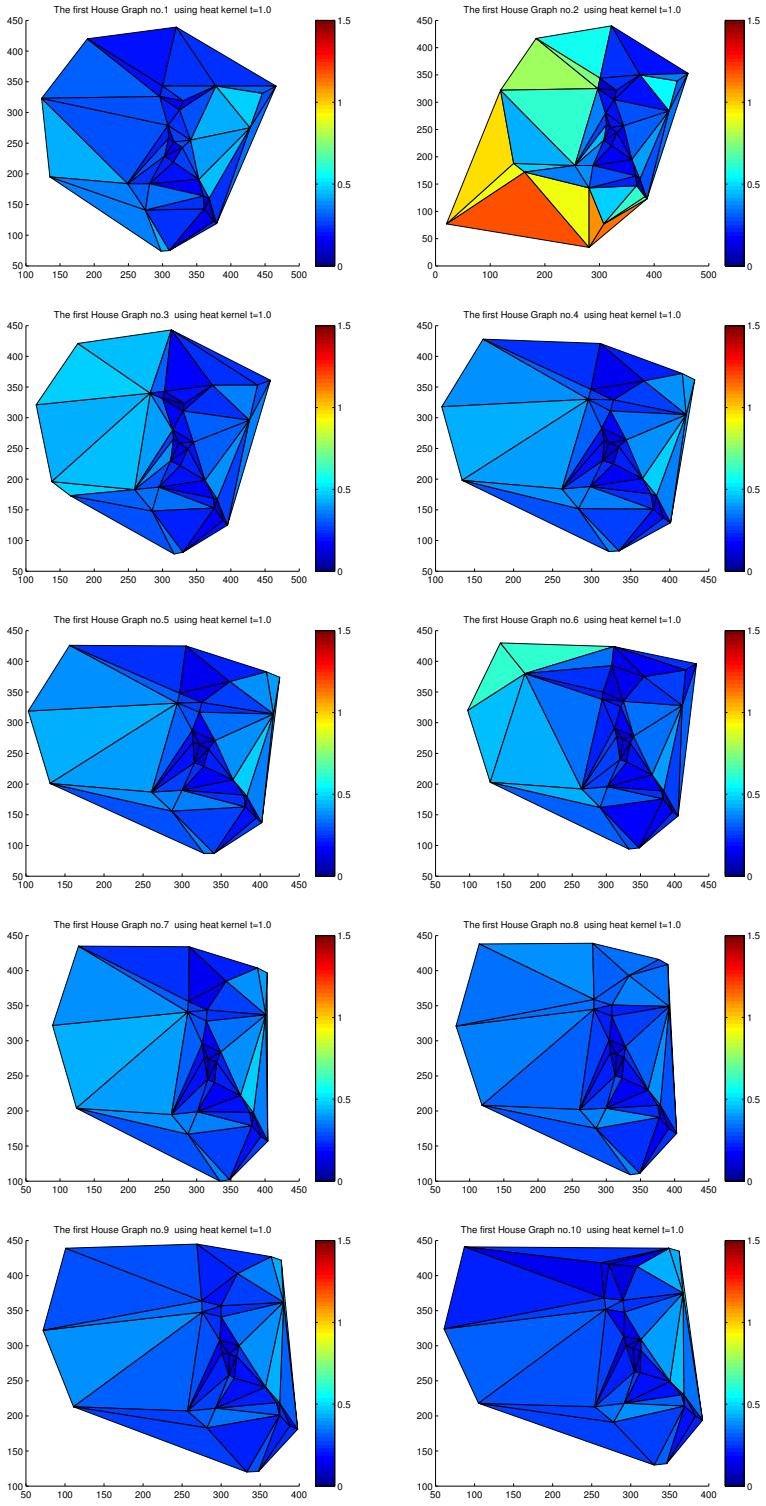


Figure 3.27: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 1st house at $t = 1.0$.

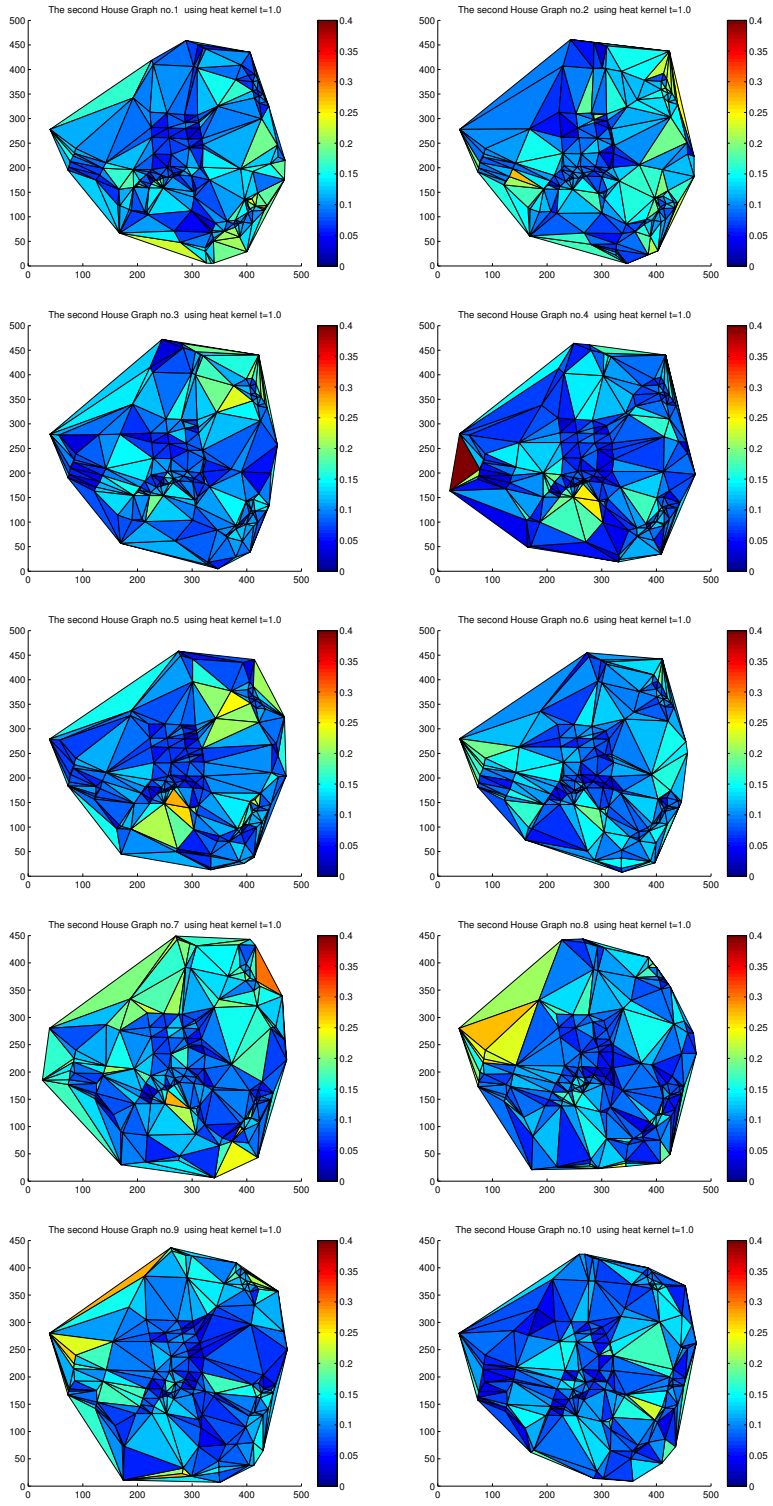


Figure 3.28: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 2nd house at $t = 1.0$.

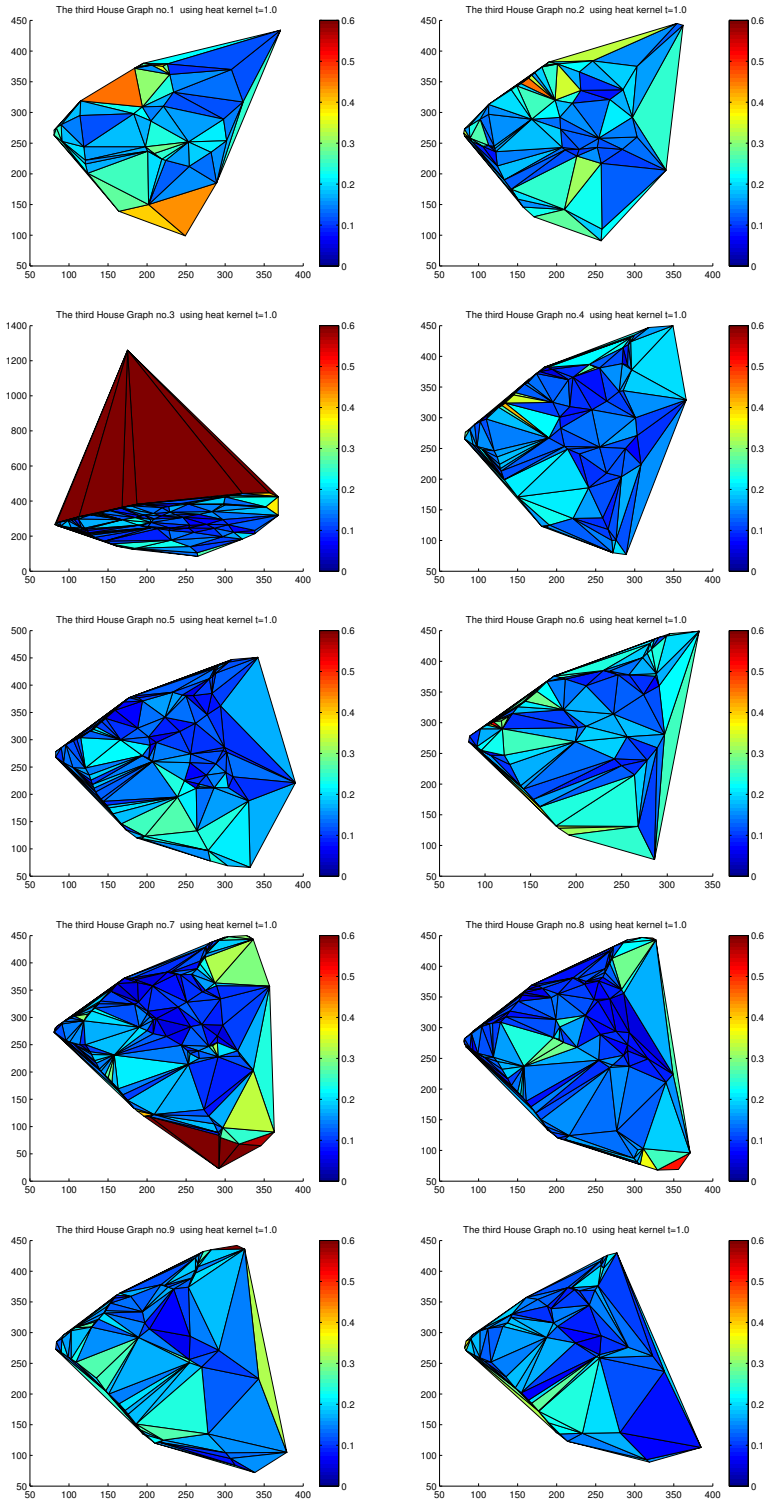


Figure 3.29: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 3rd house at $t = 1.0$.

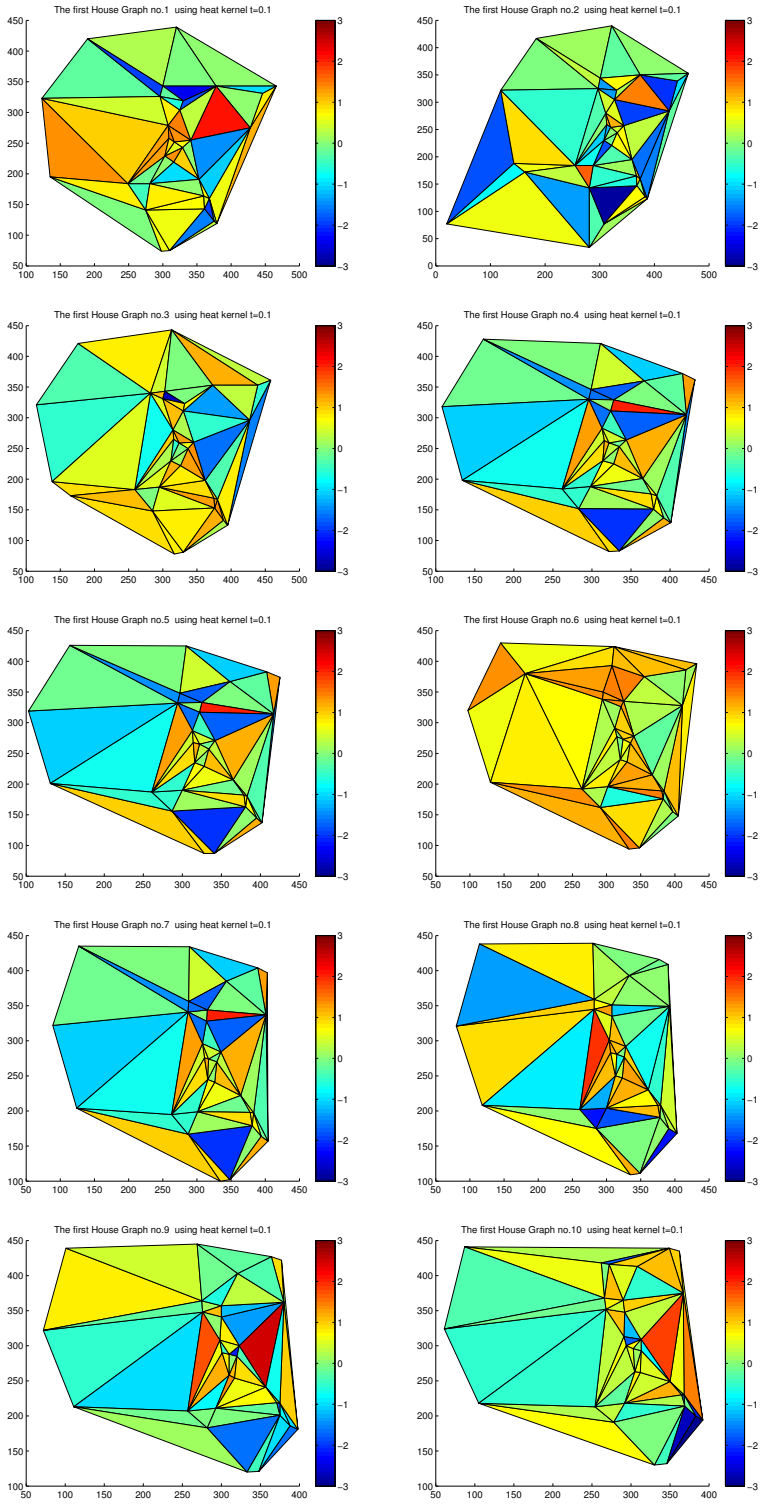


Figure 3.30: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 1st house at $t = 0.1$.

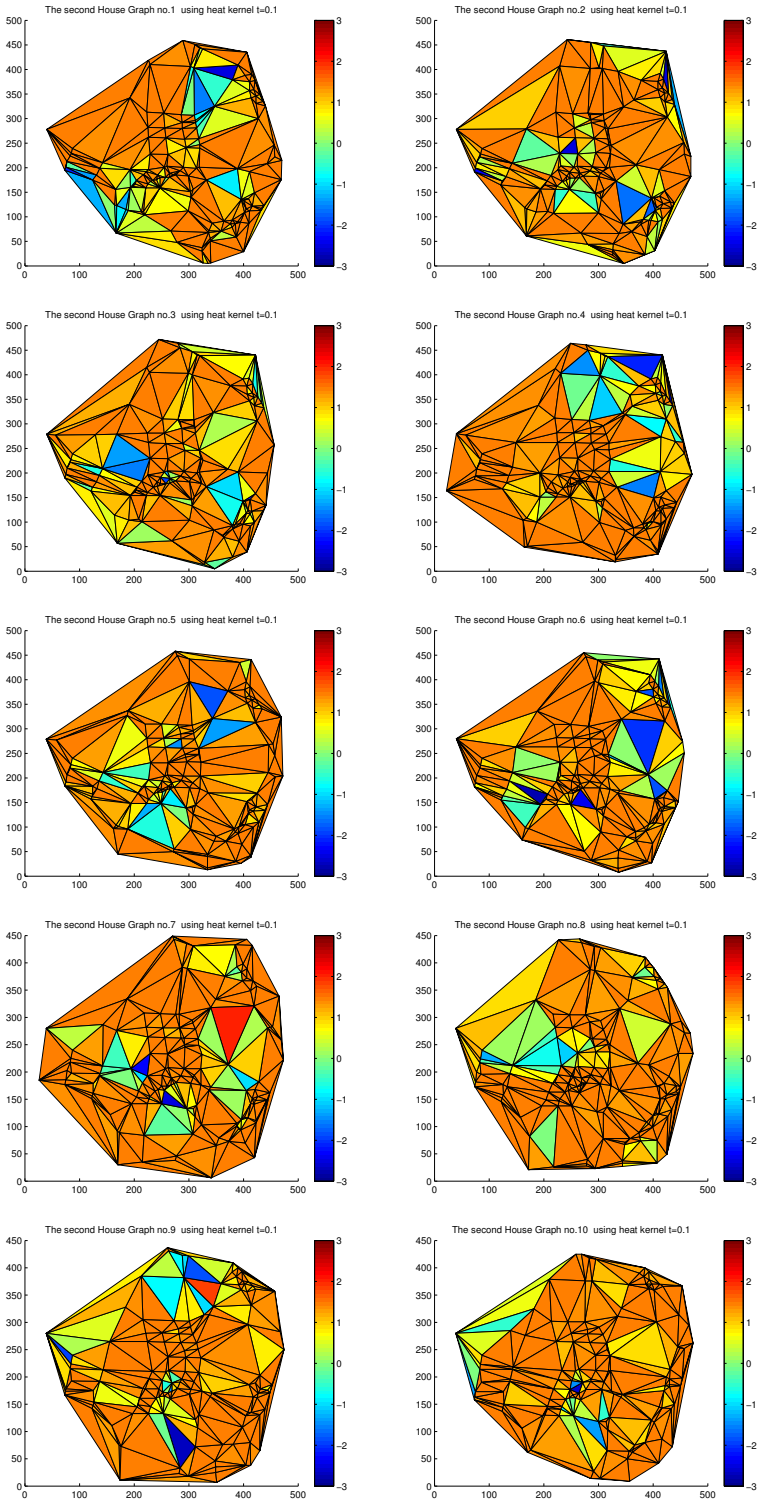


Figure 3.31: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 2nd house at $t = 0.1$.

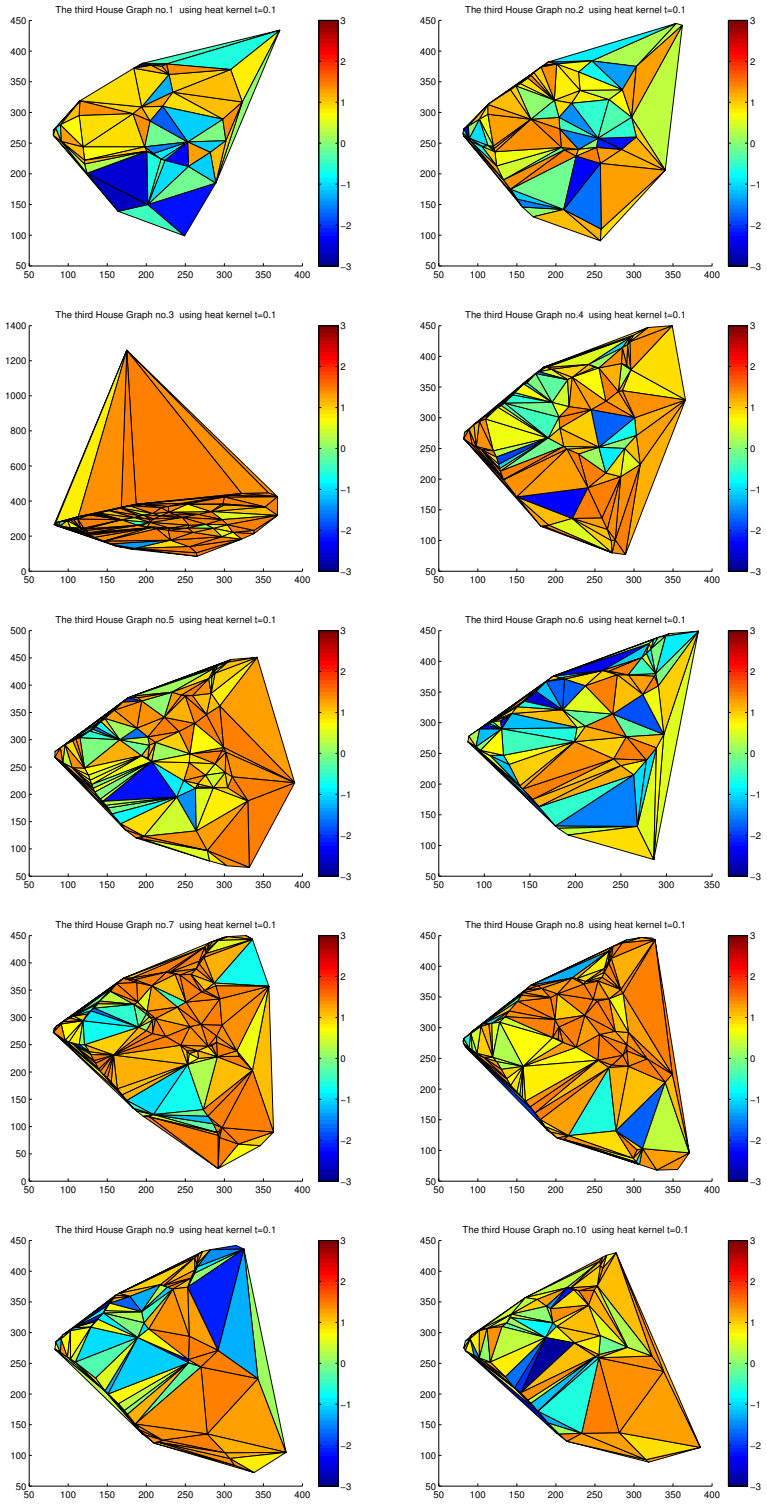


Figure 3.32: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 3rd house at $t = 0.1$.

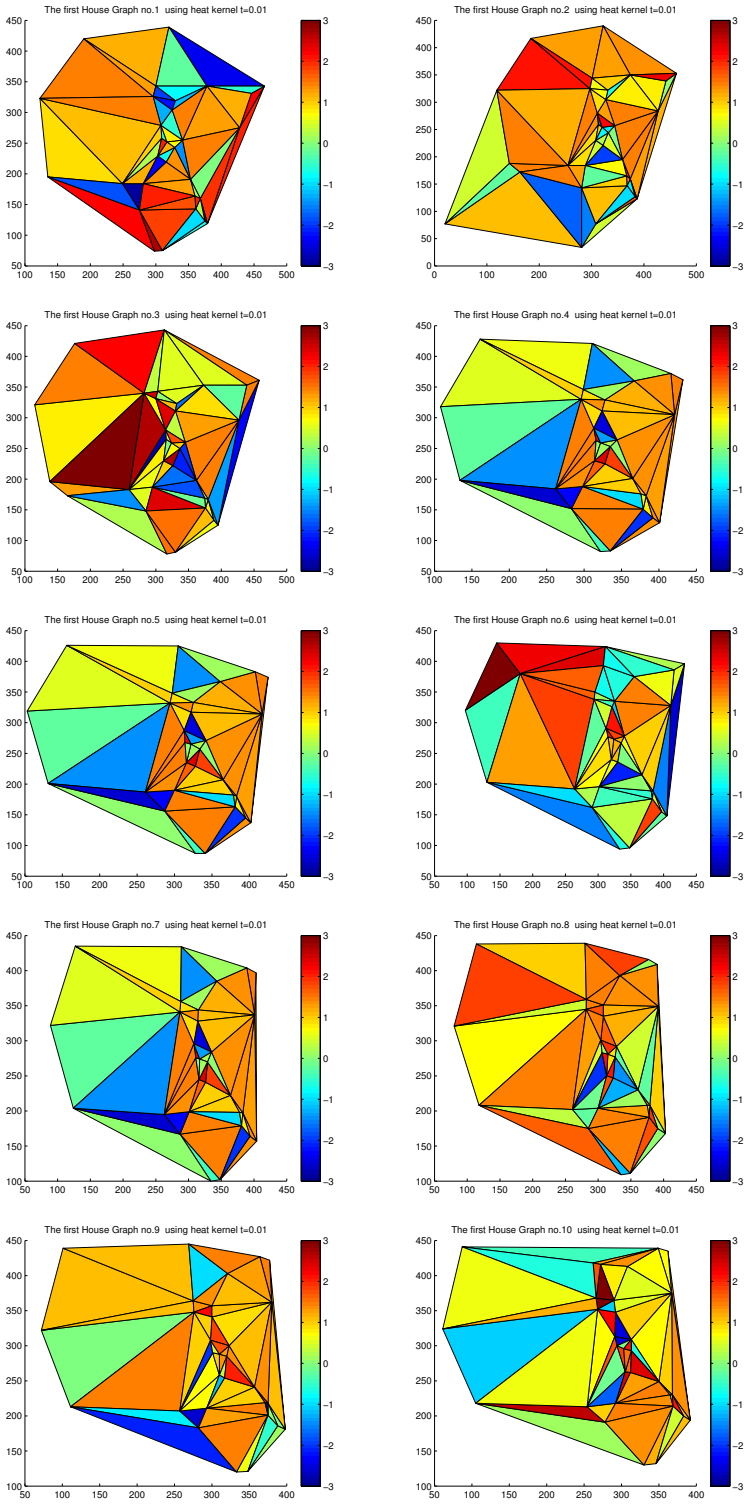


Figure 3.33: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 1st house at $t = 0.01$.

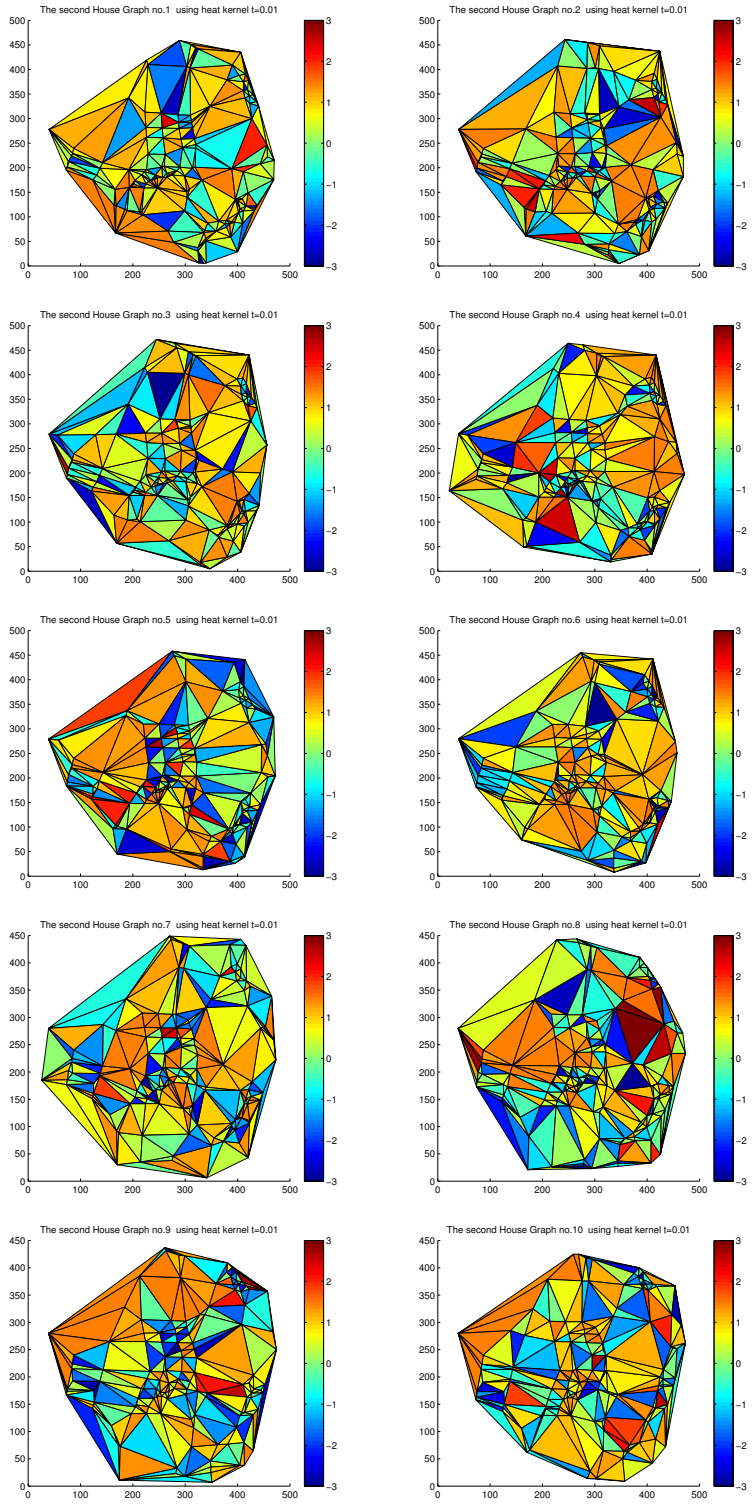


Figure 3.34: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 2nd house at $t = 0.01$.

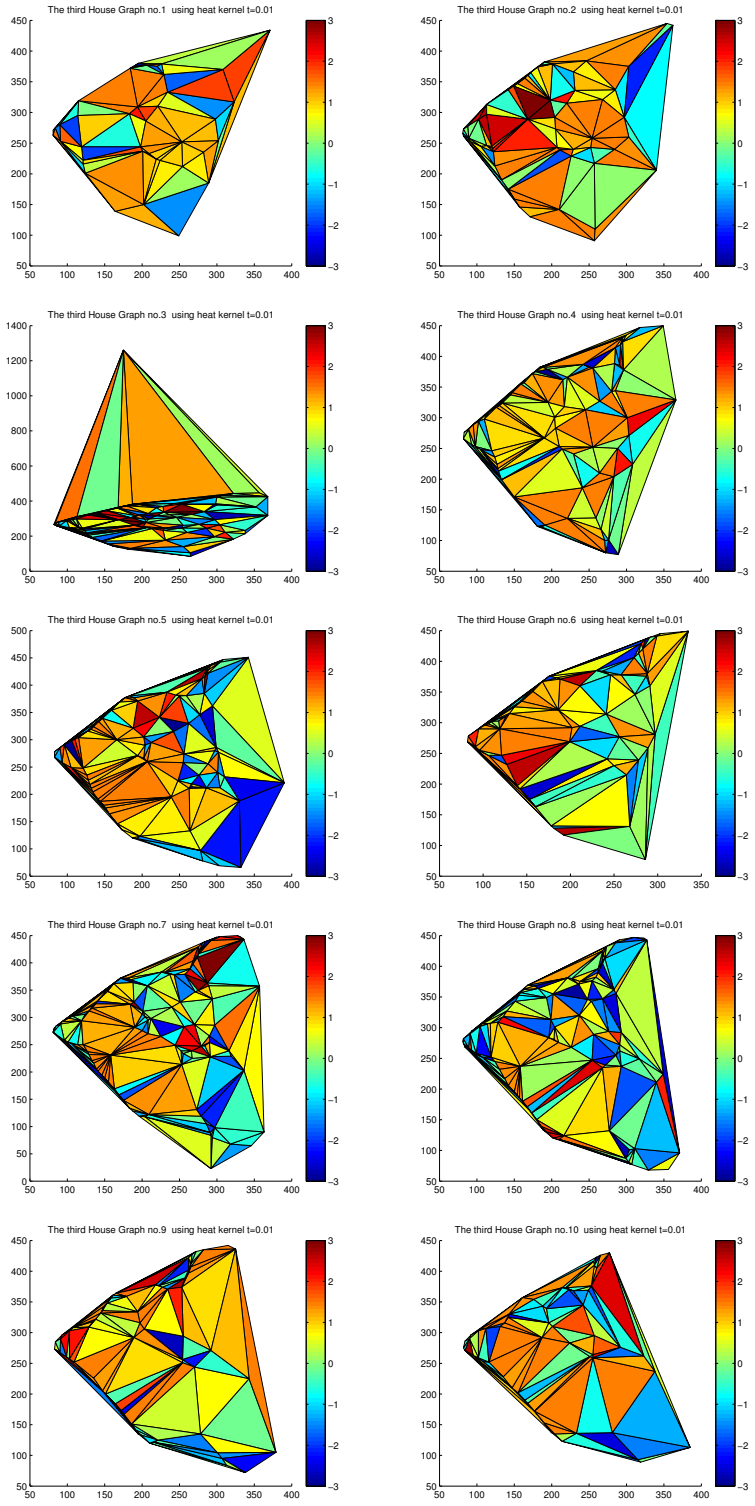


Figure 3.35: The distribution of the Gaussian curvatures of the geodesic triangles for the ten graphs of the 3rd house at $t = 0.01$.

TOY databases previously presented in Section 3.4.4. The experimental results show that the proposed characterizations (Sectional and Gaussian curvatures) might be an effective tool for clustering graphs. However, the Sectional curvature associated with the edges gives a slightly better graph clustering.

CHAPTER 4

Regularization on Graphs

4.1 Introduction

In computer vision, image processing and graphics, the data under consideration frequently exists in the form of a graph or a mesh. The fundamental problems that arise in the processing of such data are how to smooth, denoise, restore and simplify data samples over a graph. The Principal difficulty of this task, is how to preserve the geometrical structures existing in the initial data. Many methods have been proposed to solve this problem. Among existing methods, variational techniques based on regularization, provide a general framework for designing efficient filtering processes. Solutions to the variational models can be obtained by minimizing an appropriate energy function. The minimization is usually performed by designing a continuous partial differential equation, whose solutions are discretized in order to fit with the data domain. A complete overview of these methods in image processing can be found in (Bertalmio, Cheng, Osher & Sapiro, 2001; Bougleux & Elmoataz, 2005; Boykov & Huttenlocher, 1999; Chan, Osher & Shen, 2001). One of the problems associated with variational

methods is that of discretization, which for some types of data can prove to be intractable. An alternative to the variational approach, is to make direct use of differential geometry and the calculus of variation to regularize data on manifolds. There are two principal ways in which this may be effected. The first approach is to use an intrinsic-parametric description of the manifold and an explicit form of the metric, referred to as the Polyakov action (Sochen, Kimmel & Malladi, 1996, 1998; Sochen & Zeevi, 1998; Kimmel, Malladi & Sochen, 2000; Sochen & Kimmel, 2001). The second approach is to use an implicit representation of the manifold, referred to as the harmonic map (Bertalmio et al., 2001; Memoli, Sapiro & Osher, 2002; Cheng, Burchard, Merriman & Osher, 2000; Chan & Shen, 2000; Sapiro, 2001). In (Sochen, Deriche & Lopez-Perez, 2003b, 2003a, 2003c), the relation between these two approaches was explained and a new approach to perform regularization on manifolds referred to as the Beltrami flow was introduced. An implementation for the case of a manifold represented by a level set surface was introduced in (Sochen et al., 2003a). Even though the level set approach is easier to implement, for certain applications it is more convenient to handle triangulated-based techniques rather than implicit ones where the regularization process might be an intermediate step to achieve a further goal, and the same triangulation is needed for both the input and output step. A method to compute the Beltrami flow for scalar functions defined on triangulated manifolds using a local approximation of the operator was proposed in (Lopez-Perez, Deriche & Sochen, 2004). The Laplace-Beltrami operator on a Riemannian manifold has been extensively studied in the mathematics literature. In recent times, there has been intense interest in the spectral theory of the operator, which has lead to the core of the spectral geometry. This work has established relations between the first eigenvalues of the Beltrami operator and the geometrical properties of the manifold including curvatures, diameter, injectivity radius and volume. Recently, an alternative operator referred to as the p-Laplacian has attracted considerable interest, and has proved a powerful means of solving geometric nonlinear partial differential equations arising in physics.

In Chapter 3, we have explored the problem of how to characterize graphs

in a geometric manner. The idea has been to embed graphs a manifold using the heat-kernel induced from the graph. Under this embedding, nodes become points on a manifold, and each first-order cycle of the graph becomes a triangle. Using the Gauss-Bonnet theorem, we can extend this characterization to include the Gauss curvatures associated with nodes (i.e. points on a manifold) through the angular excess of the geodesic triangles.When computing the curvature characterizations we had to use some approximations to get more simpler formulae, for instance we considered the geodesic triangle to be residing on a hyper sphere as well as projecting the data onto a positive space, these approximations might led to a margin of error. For this reason in this chapter, we turn to regularization as a means of smoothing the Gaussian curvature estimates. Using the curvature information we perform regularization with the advantage of not requiring the explicit solution of a partial differential equation. To do so we investigate two cases of the p-Laplacian, the Laplace and Curvature operators, and use the Gaussian curvature associated with the heat-kernel embedding of nodes as a regularization function on the manifold. The idea of using functionals on graphs in a regularization process, has also been proposed in other contexts such as semi-supervised data learning (Zhou & Schölkopf, 2005, 2006) and image segmentation (Bougleux & Elmoataz, 2005).

In the rest of this chapter, we commence by introducing some basic functions and operators defined on graphs as discrete versions of the continuous differential ones in Section 4.2. In particular, Section 4.2.5 is devoted for a detailed overview for the p-Laplacian operator. Then a regularization framework based on the p-Laplacian operator is explained in Section 4.3. The geometric preliminaries needed to define the regularization function associated to the points on the manifold is shown in Section 4.4. To end up this chapter we introduce some experimental results and conclusions in Sections 4.6 and 4.7, respectively.

4.2 Functions and Operators on Graphs

In this section, we recall some basic prerequisites concerning weighted graphs, and define nonlocal operators which can be considered as discrete versions of continuous differential operators.

4.2.1 Preliminaries

We commence with an undirected weighted graph denoted by $G_w = (V, E)$ consists of a finite set of nodes V and a finite set of edges $E \subseteq V \times V$ represented by an $n \times n$ matrix W. Each entry ω_{uv} is the edge weight between nodes u and v, with $\omega_{uv} = 0$ if (u, v) is not in E.

$$W(u,v) = \begin{cases} \omega_{uv} & if(u,v) \in E \\ 0 & otherwise \end{cases} \tag{4.1}$$

By construction, W is symmetric and its diagonal entries are zero. As in the unweighted graph case, to construct the combinatorial graph Laplacian matrix we first establish a diagonal degree matrix D_w with elements

$$D_w(u,u) = \sum_{v \in V} \omega_{uv} = d_u \tag{4.2}$$

We then construct the Laplacian matrix $L_w = D_w - W$, that is the degree matrix minus the weight matrix.

$$L_w(u,v) = \begin{cases} d_u & if u = v \\ -\omega_{uv} & if(u,v) \in E \\ 0 & otherwise \end{cases} \tag{4.3}$$

Hence, the normalized Laplacian is given by $\widehat{L}_w = D_w^{-1/2} L_w D_w^{-1/2}$. Again, the spectral decomposition of the weighted normalized Laplacian matrix is $\widehat{L}_w = \Phi_w \Lambda_w \Phi_w^T = \sum_{i=1}^{|V|} \lambda_i \phi_i \phi_i^T$ where $|V|$ is the number of nodes and

$\Lambda_w = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|}), (0 < \lambda_1 < \lambda_2 < ... < \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues as elements and $\Phi_w = (\phi_1|\phi_2|...|\phi_{|V|})$ is the matrix with the eigenvectors as columns. A full discussion for the relation of $\widehat{L_w}$ to the weighted Laplace-Beltrami operator was introduced by Lafon in (Lafon, 2004).

4.2.2 Functions on Graphs

The connection of the Laplacian matrices described in the previous section with the theory of Regularization stands from the fact that given a real-valued function *f* defined over the vertices of G, that is, $f : V \rightarrow \Re$ assigning a real value $f(u)$ to each vertex $u \in V$, both L and $\widehat{L}$ can be described as discrete differential operators which tend to penalize changes of *f* between adjacent edges. Functions of such type form a discrete n-dimensional space. By analogy with continuous functional spaces, the discrete integral of a function $f : V \rightarrow \Re$, on the graph G, is defined by $\int_G f = \sum_{u \in V} f(u)$. Let $H(V)$ denote the Hilbert space of the real-valued functions on the vertices of G and endowed with the usual inner product:

$$\langle f,h \rangle_{H(V)} = \sum_{u \in V} f(u)h(u) \quad , \quad f,h : V \rightarrow \Re \tag{4.4}$$

with the induced L_2- norm: $\|f\|_2 = \langle f,f \rangle_{H(V)}^{1/2}$.
In a similar way we can define $H(E)$, the space of real-valued functions on edges, endowed with the inner product

$$\langle F,H \rangle_{H(E)} = \sum_{v \in V} \sum_{(u,v) \in E} F(u,v)H(u,v), \quad , \quad F, H : E \rightarrow \Re \tag{4.5}$$

We can say that functions in $H(E)$ do not need to be symmetric, and their inner product can be rewritten as:

$$\langle F,H \rangle_{H(E)} = \sum_{(u,v) \in E} F(u,v)H(u,v), \quad , \quad F, H : E \rightarrow \Re \tag{4.6}$$

and the induced L_2- norm is defined by: $\|F\|_2 = \langle F,F\rangle_{H(E)}^{1/2}$.

4.2.3 Regularization by means of the Laplacian

If we put f in a column vector form, that is $f \in \Re^n$, the following inner product measures the smoothness of f over the graph G

$$\langle f, \widehat{L}f \rangle = f^T \widehat{L} f = \frac{1}{2} \sum_{(u,v)\in E} \omega_{uv}(f(u) - f(v))^2 \geqslant 0 \tag{4.7}$$

that is a smaller value means smoother f. Roughly speaking, f is smooth if $f(u) \approx f(v)$ for those pairs with large ω_{uv}. This is sometimes informally expressed by saying that f varies slowly over the graph, or that f follows the data manifold. Note that

$$f^T \widehat{L} f = \frac{1}{2} \sum_{(u,v)\in E} \omega_{uv}(f(u) - f(v))^2 \geqslant 0 \tag{4.8}$$

where the inequality holds because W has non-negative entries.

Alternatively a way of formulating regularization through spectral analysis was introduced by Smola et al (Smola, Schölkopf & Müller, 1998; Smola & Kondor, 2003b), where they suggest a spectral-based regularization that comes from

$$\langle f, \widehat{L}f \rangle = f^T [\sum_{i=1}^{n} \lambda_i \phi_i \phi_i^T] f = \sum_{i=1}^{n} \langle f, \phi_i \rangle \lambda_i \langle \phi_i, f \rangle \tag{4.9}$$

In particular, the smoothness of an eigenvector is

$$\phi_i^T \widehat{L} \phi_i = \lambda_i \tag{4.10}$$

Hence, eigenvectors with smaller eigenvalues are smoother. Since $\{\phi_i\}$ forms a basis on $\Re^n$, we can write any function f as

$$f = \sum_{v \sim u}^{n} a_i \phi_i \quad , a_i \in \Re \tag{4.11}$$

and equation (4.9) which measures the smoothness of f can be re-expressed as

$$\langle f, Lf \rangle = \sum_{v \sim u}^{n} a_i^2 \lambda_i \tag{4.12}$$

Finally, the idea was extended to a class of regularization functionals on graphs in the following sense

$$\langle f, Pf \rangle = \langle f, p(\widehat{L}) f \rangle \tag{4.13}$$

Here, $p(\widehat{L})$ is simply considered to be as applying the scalar valued function $p(\lambda)$ to the eigenvalues of $\widehat{L}$, that is

$$p(L) = \sum_{i=1}^{n} p(\lambda_i) \phi_i \phi_i^T \tag{4.14}$$

where (λ_i, ϕ_i) is the eigensystem of $\widehat{L}$.

The following functions, $p(\lambda_i)$, are of particular interest where $i = 1, 2, ..., |V|$:
Regularized Laplacian :

$$p(\lambda_i) = 1 + \sigma^2 \lambda_i \tag{4.15}$$

Diffusion Process :

$$p(\lambda_i) = exp(\sigma^2 / 2\lambda_i) \tag{4.16}$$

One-Step Random Walk :

$$p(\lambda_i) = (\alpha - \lambda_i)^{-1} \quad , \alpha \geq 2 \tag{4.17}$$

p-Step Random Walk :

$$p(\lambda_i) = (\alpha - \lambda_i)^{-p} \quad , \alpha \geq 2 \tag{4.18}$$

Inverse Cosine :

$$p(\lambda_i) = (\cos \lambda_i \pi/4)^{-1} \tag{4.19}$$

Using the definition (4.11) of a function f we can rewrite equation (4.13) as follows

$$\langle f, p(\widehat{L})f \rangle = \sum_{v \sim u}^{n} a_i^2 p(\lambda_i) \tag{4.20}$$

4.2.4 Operators on Graphs

The difference operator $d : H(V) \rightarrow H(E)$ of a function $f \in H(V)$ on an edge $(u, v) \in E$, is defined by

$$d_f = \sqrt{\omega_{uv}}(f(v) - f(u)) \ , \qquad \forall (u, v) \in E \tag{4.21}$$

The directional derivative (or edge derivative) of a function $f \in H(V)$ at a vertex v along an edge (u, v), is defined as $\partial_v f_u = d_f(u, v)$. This definition is consistent with the continuous definition of the derivative of a function, that is if $f(v) = f(u)$ then $\partial_v f_u = 0$. Moreover, one has $\partial_v f_u = -\partial_u f_v$ and $\partial_v f_v = 0$. the gradient operator ∇ of a function $f \in H(V)$ at a vertex v is the vector operator defined by $\nabla f(v) = (\partial_v f_u)^T$, for $(u, v) \in E$. The local variation of f at v, is defined to be

$$\|\nabla f(v)\| = \sqrt{\sum_{(u,v) \in E} (\partial_v f_u))^2} = \sqrt{\sum_{(u,v) \in E} \omega_{uv}(f(u) - f(v))^2} \tag{4.22}$$

which can be viewed as a measure of the regularity of a function around a vertex.

4.2.5 The *p*-Laplacian Operator

The p-Laplace operator describes a family of second order operators. For a smooth Riemannian manifold M and a real number $p \in (1, +\infty)$, the p-Laplacian operator of a function $f \in H(V)$, denoted $L^p : H(V) \to H(V)$ is defined by

$$L^p f(u) = \frac{1}{p} \sum_{(u,v)\in E} \omega_{uv}(\|\nabla f(u)\|^{p-2} + \|\nabla f(v)\|^{p-2})(f(u) - f(v)) \tag{4.23}$$

This operator arises naturally from the variational problem associated to the energy function (Lim, Montenegro & Santos, 2008). The p-Laplace operator is nonlinear, with the exception of $p = 2$, where it corresponds to the combinatorial graph Laplacian, which is one of the classical second order operators defined in the context of spectral graph theory (Chung, 1997)

$$Lf(u) = \sum_{(u,v)\in E} \omega_{uv}(f(u) - f(v)) \tag{4.24}$$

Another particular case of the p-Laplace operator is obtained with p=1 . In this case, it is the curvature of the function f on the graph

$$\kappa f(u) = \frac{1}{2} \sum_{(u,v)\in E} \omega_{uv}\left(\frac{1}{\|\nabla f(v)\|} + \frac{1}{\|\nabla f(u)\|}\right)(f(u) - f(v)) \tag{4.25}$$

κ corresponds to the curvature operator proposed in (Osher & Shen, 2000) and (Chan et al., 2001) in the context of image restoration. More generally, κ is the discrete analogue of the mean curvature of the level curve of a function defined on a continuous domain of $\Re^N$.

In the case when $\omega_{uv} = 1$ the formulas 4.23, 4.24 and 4.25 become

$$L^p f(u) = \frac{1}{2} \sum_{(u,v)\in E} \left(\frac{f(u) - f(v)}{\left(\sqrt{\sum_{(v,u)\in E}(f(u) - f(v))^2}\right)^{p-2}} + \frac{f(u) - f(v)}{\left(\sqrt{\sum_{(w,v)\in E}(f(v) - f(w))^2}\right)^{p-2}}\right) \tag{4.26}$$

$$Lf(u) = \sum_{(u,v)\in E} (f(u) - f(v)) \tag{4.27}$$

and

$$\kappa f(u) = \frac{1}{2} \sum_{(u,v)\in E} \left(\frac{f(u) - f(v)}{\sqrt{\sum_{(v,u)\in E}(f(u) - f(v))^2}} + \frac{f(u) - f(v)}{\sqrt{\sum_{(w,v)\in E}(f(v) - f(w))^2}}\right) \tag{4.28}$$

In the formulae (4.26) and (4.28), the computations not only moves from the point u to its neighbouring nodes v but it takes that one more step to those nodes, w, connected to v. There is much literature on the p-Laplacian in the continuous case (Heinonen, Kilpelainen & Martio, 1993) beside some work done on discrete analogue of the p-Laplacian (Yamasaki, 1986).

4.3 *p*-Laplacian regularization framework

In a real world data, a given function $f \in H(V)$ defined on the vertices of a weighted graph $G = (V, E, W)$ usually corrupted by a noise. To recover the uncorrupted function $f^* \in H(V)$ which is smooth enough on G, and also close enough to f (Zhou & Schölkopf, 2004). This problem can be formalized by considering the optimization problem

$$f^* = \min_{f\in H(V)} \delta_w^p(f) + \frac{\lambda}{2}\|Lf\|^2 \tag{4.29}$$

which typically involves a regularization term (the first part in formula 4.29)

that measures the smoothness of the function f plus an approximation term (the second part in formula **??**) which measures the closeness to the given function f, The deal between these two terms is captured by a non negative parameter, $\lambda \geq 0$, hence by varying the value of λ, the variational problem 4.29 allows to describe the function f at different scales (each scale corresponding to a value of λ). Additionally, the regularization functional $\delta_w^p \rightarrow \Re$ is given as

$$\delta_w^p = \frac{1}{p}\sum_{u\in V} \|\nabla_w f(v)\|^p = \frac{1}{p}\sum_{u\in V}(\sum_{(v,u)\in E} \omega_{uv}(f(u) - f(v))^2)^{\frac{1}{p}} \tag{4.30}$$

where the degree of regularity, which has to be preserved, is controlled by the value of p.

In general, the proposed optimization problem 4.29 can be seen as an extension of the two classical cases of $p = 2$ or $p = 1$. In (Elmoataz, Lezoray & Bougleux, 2008), the authors showed that it has a unique solution.

4.4 The Gaussian Curvature

Curvature, is a local measure of geometry and can be used to represent local shape information. We choose the function f to be the Gaussian curvature defined over the vertices. Gaussian curvature is one of the fundamental second order geometric properties of a surface, and it is an intrinsic property of a surface independent of the coordinate system used to describe it. As stated by Gauss's theorema egregium (Gauss, 1900), it depends only on how distance is measured on the surface, not on the way it is embedded on the space.

4.4.1 Geometric Preliminaries

Let T be the embedding of a triangulated graph onto a smooth surface M in $\Re^3$, A_G be the area of a geodesic triangle on M with angles $\{\alpha_i\}_{i=1}^3$ and geodesic edge lengths $\{d_{Gi}\}_{i=1}^3$, and A_E be the area of the corresponding Euclidean triangle with edge lengths $\{d_{Ei}\}_{i=1}^3$ and angles $\{\varphi_i\}_{i=1}^3$. Assuming that each geodesic

is a great arc on a sphere with radius r_i, $i = 1, 2, 3$ corresponding to a central angle 2θ, and that the geodesic triangle is a triangle on the surface of a sphere with radius $r = \frac{1}{3}\sum_{i=1}^{3} r_i$, with the Euclidean distance between the pair of nodes to be $d_E = \frac{1}{3}\sum_{i=1}^{3} d_{Ei}$. Considering a small area element on the sphere given in spherical coordinates by $dA = r^2 \sin\theta d\theta d\varphi$, the integration of dA bounded by 2θ gives us the following formula for computing the area of the geodesic triangle

$$A_G = \frac{1}{2r} d_E^2 \tag{4.31}$$

where d_E^2 is computed from the embedding using (3.14).

4.4.2 Gaussian Curvature from Gauss Bonnet Theorem

For a smooth compact oriented Riemannian 2-manifold M, let $\triangle_G$ be a triangle on M whose sides are geodesics, i.e. paths of shortest length on the manifold. Further, let α_1, α_2 and α_3 denote the interior angles of the triangle. According to Gauss's theorem, if the Gaussian curvature K (i.e. the product of the maximum and minimum curvatures at a point on the manifold) is integrated over $\triangle_G$, then

$$\int_{\triangle_G} K dM = \sum_{i=1}^{3} \alpha_i - \pi \tag{4.32}$$

where dM is the Riemannian volume element. Since all the points, except for the vertices, of a piecewise linear surface have a neighborhood isometric to a planar Euclidean domain with zero curvature, the Gauss curvature is concentrated at the isolated vertices. Hence, to estimate the Gaussian curvature of a smooth surface from its triangulation, we need to normalize by the surface area, which here is the area of the triangle. Consequently, we will assign one third of the triangle area to each vertex. Hence, the Gaussian curvature associated with each vertex will be

$$\kappa_g = \frac{\int_{\triangle_G} K dM}{\frac{1}{3} A_G} \tag{4.33}$$

from (4.32) we get

$$\kappa_g = \frac{\sum_{i=1}^{3} \alpha_i - \pi}{\frac{1}{3} A_G} \tag{4.34}$$

From (3.30) we know that $\sum_{i=1}^{3} \alpha_i - \pi = \frac{A_G}{r^2}$. Hence, substituting in (4.34) we get

$$\kappa_g = \frac{3}{r^2} \tag{4.35}$$

Recalling that the Gaussian curvature is the product of the two principle curvatures, and that the curvature of a point on a sphere is the reciprocal of the radius of the sphere gives an explanation for the result in (4.35). As we assumed earlier that the geodesic is a great arc of a circle of radius r, in Section 3.3.1 we deduced that

$$\frac{1}{r^2} = d_G - \frac{24(d_G - d_E)}{d_G^3} \tag{4.36}$$

and since for an edge of the graph $d_G = 1$, we have

$$\frac{1}{r^2} = 24(1 - d_E) \tag{4.37}$$

From (4.35) and (4.37) the Gaussian curvature associated with the embedded node can be found from the following formula

$$\kappa_g = 72(1 - d_E) \tag{4.38}$$

4.5 The Euler characteristic

To estimate a global topological characteristic for the manifold, we can use the Gauss Bonnet Theorem stated previously in Section 3.3. We commence by triangulating M so that each face is a geodesic triangle.

Summing (4.32) over all the triangular faces gives that the integral of K over

all M is 2π times the Euler characteristic [1] of M, i.e.

$$\int_M KdM = 2\pi\chi(M) \tag{4.39}$$

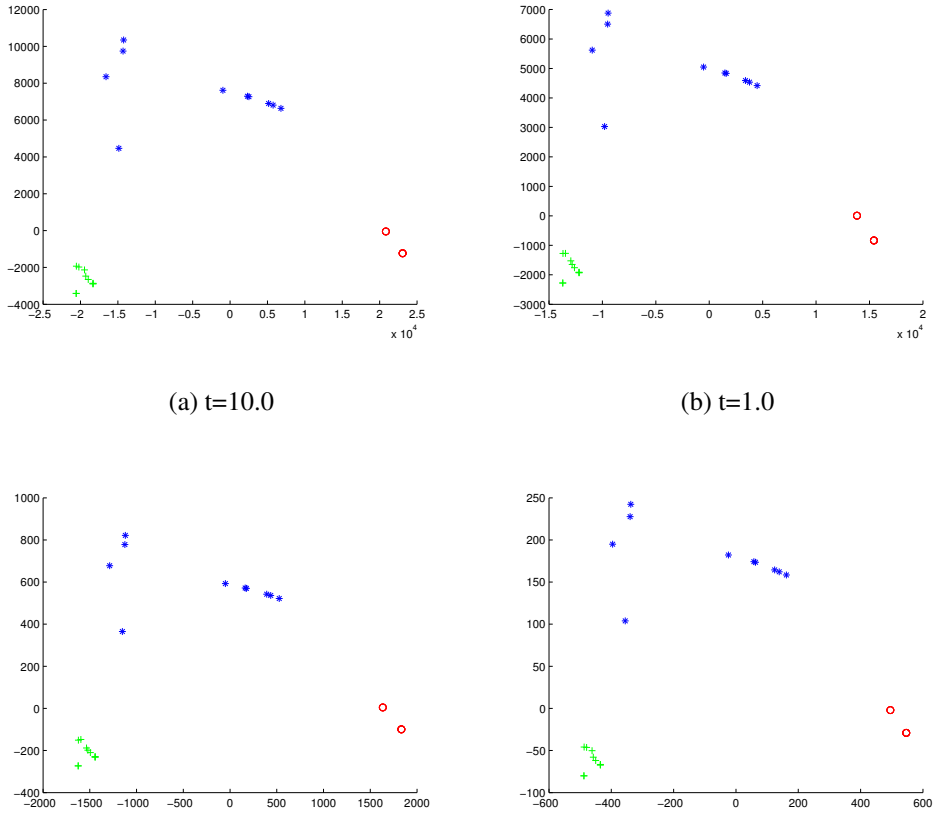
4.6 Experiments

To commence, we compute the Euclidean distances between the nodes in each graph based on the Laplacian and then on the heat kernel with the values of $t = 10.0, 1.0, 0.1$ and 0.01. Then we compute the Gaussian curvature associated with each node using the formula given in Section 4.4.2.

Commencing with each node attributed with the the Gaussian curvatures (as the value of a real function f acting on the nodes of the graph), we can regularize each graph by applying the the *p*-Laplacian operator to the Gaussian curvatures. For each graph we construct a set of regularized Gaussian curvatures using both the Laplace operator and the curvature operator, as a special cases of the *p*-Laplacian operator. The next step is to compute the distances between the sets for the different graphs using the classical Hausdorff distance and the modified Hausdorff distance. Finally, we subject the distance matrices to the Multidimensional Scaling (MDS) procedure to embed them into a 2D space. Here each graph is represented by a single point. We commence by introducing the results obtained when experimenting with the York model house database. Where each graph of the CMU model house sequence is represented as a red circle and each graph of the MOVI model house sequence is represented as blue star while each graph of the Swiss chalet model house sequence is represented as a green cross. Figures 4.1 and 4.2, show the results when using the Hausdorff distance (HD) to measure the (dis)similarity between pairs of graphs regularized by using the Laplace and Curvature operators respectively. The subfigures are ordered from left to right, top to bottom using the heat kernel embedding with the values $t = 10.0, 1.0, 0.1$ and 0.01 respectively. With the same order, Figures 4.3

[1] The Euler characteristic is a topological invariant, a number that describes a topological space's shape or structure regardless of its geometry (Early, 1999).

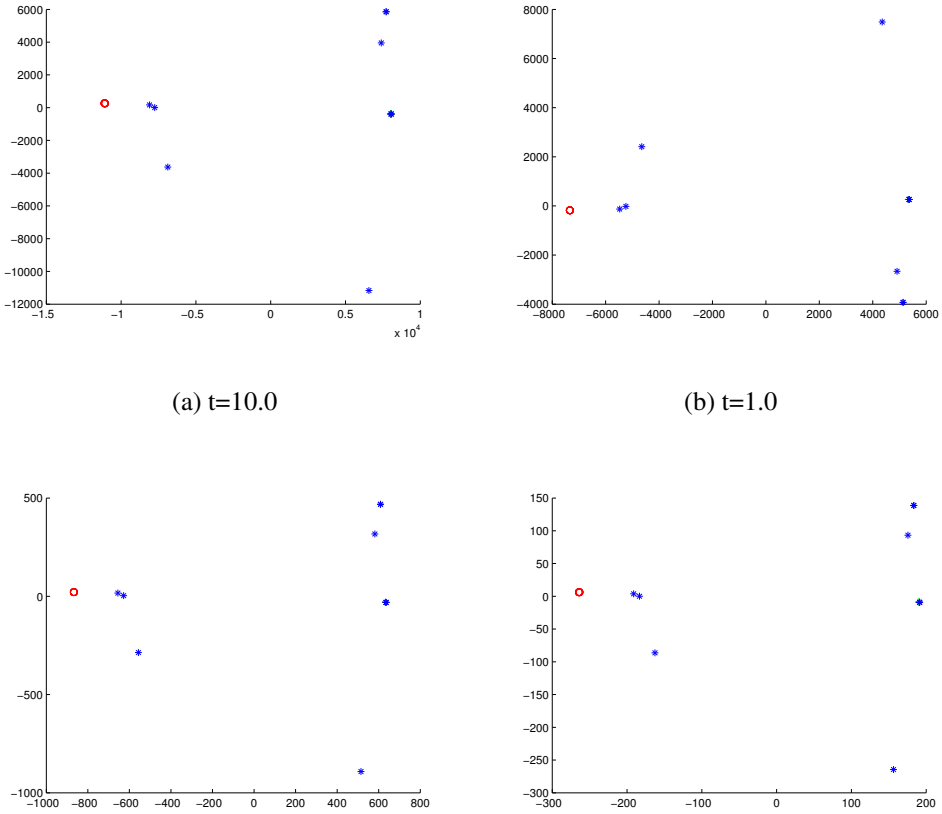
and 4.4, give the results obtained when using the Modified Hausdorff distance (MHD).



(a) t=10.0 (b) t=1.0

(c) t=0.1 (d) t=0.01

Figure 4.1: MDS embedding obtained using HD when apply Laplace operator to regularize the houses data resulting from the heat kernel embedding.



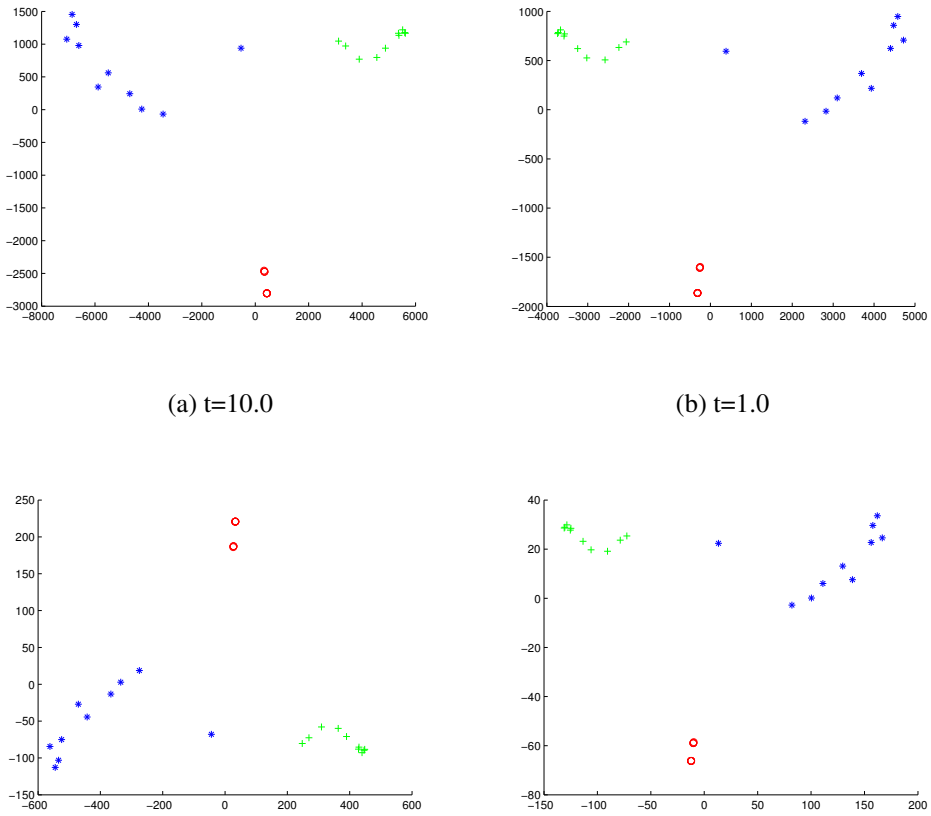
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 4.2: MDS embedding obtained using HD when apply Curvature operator to regularize houses data resulting from the heat kernel embedding.



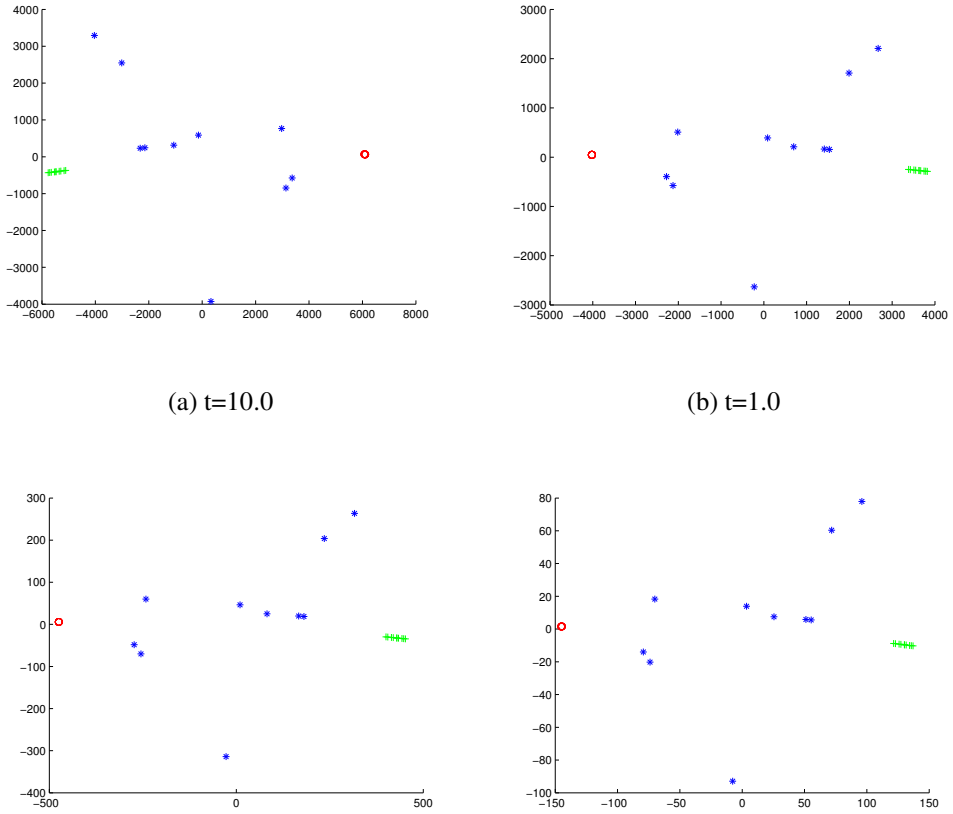
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 4.3: MDS embedding obtained using MHD when apply Laplace operator to regularize the houses data resulting from the heat kernel embedding.



(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 4.4: MDS embedding obtained using MHD when apply Curvature operator to regularize houses data resulting from the heat kernel embedding.

		t=10	t=1.0	t=0.1	t=0.01
HD	Laplace operator	0.0667	0.0667	0.0667	0.0667
HD	Curvature operator	0.3333	0.3333	0.3333	0.3333
MHD	Laplace operator	0.0333	0.0333	0.0333	0.0333
MHD	Curvature operator	0.1333	0.1333	0.1333	0.1333

Table 4.1: A rand index vs. t. for the houses dataset

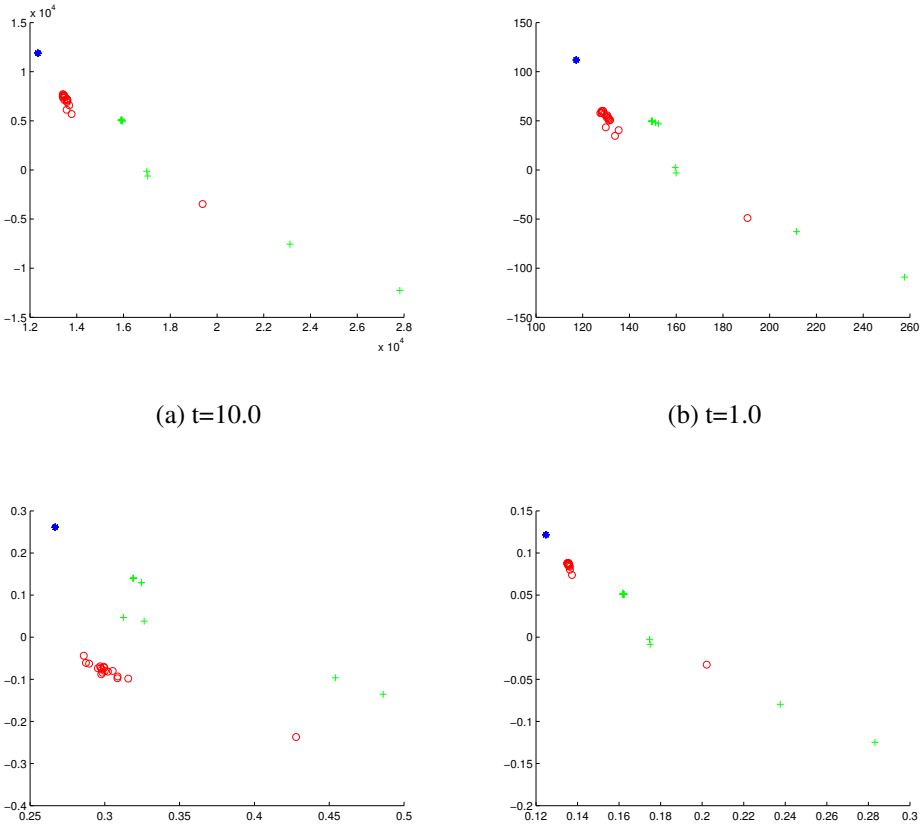
To investigate the results in more details table 4.1 shows the rand index for the distance as a function of t. This index is computed as explained in Section 3.4.5.

Next, we consider the Euler characteristic whose computation was detailed in Section 4.5. In Table 4.6 we list the mean and variance for each group of graphs (for each house).

		t=10	t=1.0	t=0.1	t=0.01
1st house	Mean	-0.1725	2.2361	1.2835	5.1177
	Variance	0.0001	0.1830	4.8437	1.7395
2nd house	Mean	-0.7514	4.0221	49.1613	8.3172
	Variance	0.0003	0.1077	12.4022	10.2801
3rd house	Mean	-0.3970	3.5168	16.1415	6.1704
	Variance	0.0120	0.8116	150.9340	17.5878

Table 4.2: The mean and variance for the Euler characteristic of the manifold embedding of each graph.

Now, we give the results obtained when experimenting with the COIL database. Where each graph of the sequence of the first object is represented as a red circle and each graph of the sequence of the second object is represented as blue star while each graph of the sequence of the third object is represented as a green cross. Figures 4.5 and 4.6 show the results when using the Hausdorff distance (HD) to measure the (dis)similarity between pairs of graphs regularized by using the Laplace and Curvature operators respectively. The subfigures are ordered from left to right, top to bottom using the heat kernel embedding with the values $t = 10.0, 1.0, 0.1$ and 0.01 respectively. With the same order, Figures 4.7 and 4.8 give the results obtained when using the Modified Hausdorff distance (MHD). Followed by Table 4.3 giving the Rand Index for the data as a function of t.



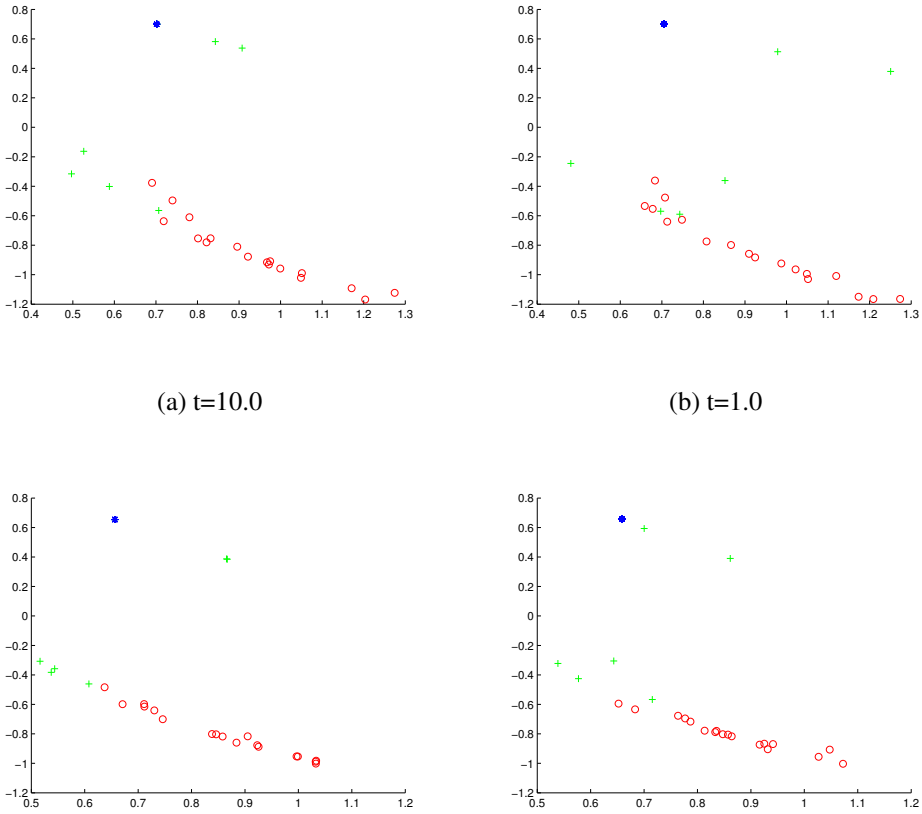
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

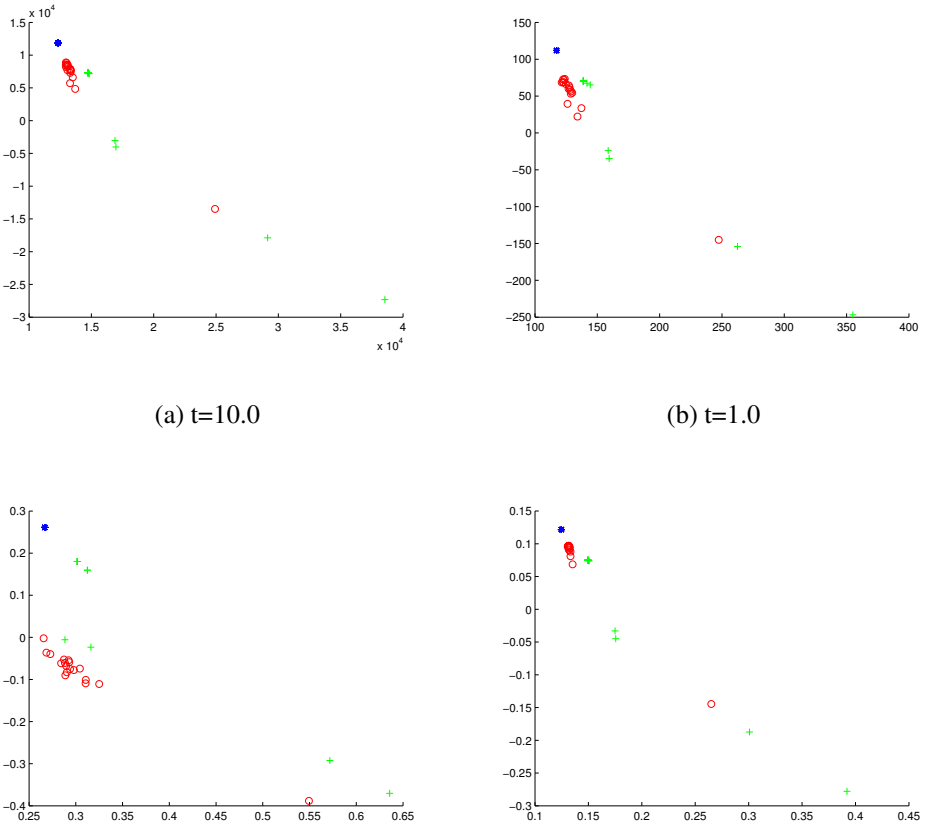
Figure 4.5: MDS embedding obtained using HD when apply Laplace operator to regularize the COIL data resulting from the heat kernel embedding.



(a) t=10.0 (b) t=1.0

(c) t=0.1 (d) t=0.01

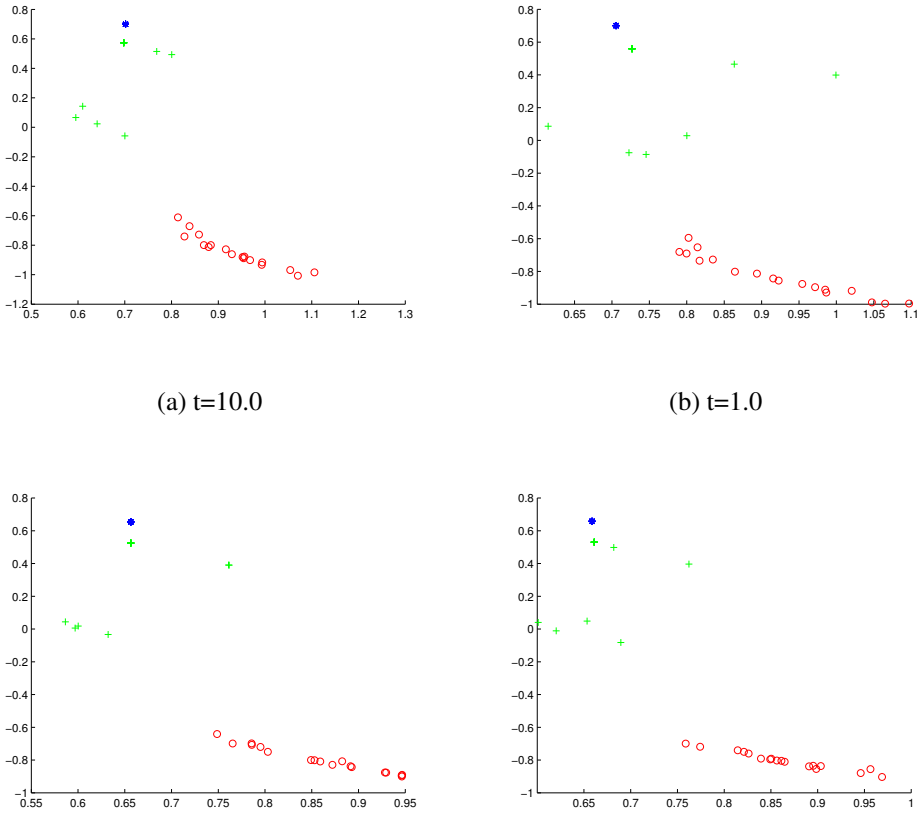
Figure 4.6: MDS embedding obtained using HD when apply Curvature operator to regularize COIL data resulting from the heat kernel embedding.



(a) t=10.0 (b) t=1.0

(c) t=0.1 (d) t=0.01

Figure 4.7: MDS embedding obtained using MHD when apply Laplace operator to regularize the COIL data resulting from the heat kernel embedding.



(a) t=10.0 (b) t=1.0

(c) t=0.1 (d) t=0.01

Figure 4.8: MDS embedding obtained using MHD when apply Curvature operator to regularize COIL data resulting from the heat kernel embedding.

		t=10	t=1.0	t=0.1	t=0.01
HD	Laplace operator	0.2407	0.2778	0.0370	0.0185
HD	Curvature operator	0.2963	0.2778	0.2963	0.3148
MHD	Laplace operator	0.2778	0.2778	0.0741	0.2778
MHD	Curvature operator	0.2222	0.0185	0.0000	0.2222

Table 4.3: A rand index vs. t. for the COIL dataset

From this experimental study, we can figure out a number of observations. For instance, when experimenting with the York model house's dataset, the MDS results produced by both the HD and MHD distance measures give a clear clusters for the three groups even though the clusters of the MOVI model house's sequence is less compact than the other two clusters (CMU and Swiss chalet model houses' sequences). Moreover, the CMU model house sequence is embedded around two points and sometimes around one point only. Furthermore, Table (4.1) shows that the error is less when experimenting the Laplacian operator using the MHD distance measure.

To understand these results, we refer to Tables 3.1, 3.2 and 3.3, which show that the images' sequences of the MOVI model house contain more feature points and little texture than the CMU and Swiss chalet model houses' sequences. Furthermore, the MOVI sequence has a wider range of the node, edge and face frequencies as one can see from Figures 3.4, 3.5 and 3.6.

As for the COIL data, the clusters were clear when experimenting with Curvature operator using the MHD distance measure. These results might be confirmed form table (4.3) which shows that the error deduced in this situation is as small as zero up to four digits specially at values of t equal to 1.0 and 0.1. However, the graphs of the second object are embedded around a single point in all the cases.

4.7 Conclusion

Based on the p-Laplacian operators, we proposed a framework to regularize real-valued or vector-valued functions on weighted graphs of arbitrary topology. The

approach considers minimizing a weighted sum of two energy terms: a regularization one that uses the discrete p-Dirichlet form, and an additional approximation one which helps to avoid the shrinkage effects obtained during the regularization process with appropriate choice for the regularization parameter λ. The proposed model is parameterized by the degree p of regularity, the graph structure and the weight function. The data can be structured by functions depending on data features, the curvature attributes associated with the geometric embedding of the graph.

The proposed framework brings together several distinct concepts that have recently received some significant attention in machine learning. Firstly, the techniques deduced from the spectral graph theory (Chung, 1997) that has been applied to a wide range of clustering and classification tasks over the last decades. Taking in consideration the properties of the graph p-Laplacian as a nonlinear extension of the usual graph Laplacian. Secondly, the geometric point of view which comes from the heat kernel embedding of the graph into a manifold in a class of algorithms that can be termed as manifold learning. In these techniques, we use the geometry of the manifold by assuming that it has the geometric structure of a Riemannian manifold. Thirdly, the conceptual framework which comes from the manifold regularization and extends the classical framework of regularization in the sense of reproducing Hilbert Spaces to exploit the geometry of the embedded set of points.

In the section of experiments, results are provided for both the Laplace and Curvature operators as a two special cases for the p-Laplacian regularization framework. The two techniques were used for gauging the graph similarity. The databases used for that purpose were the York model house and some selected items from the COIL-20 database previously presented in Section 3.4.4. Experiments show that it is an efficient procedure for the purpose of gauging the similarity of pairs of graphs. With an appropriate choice for the value of $\lambda > 0$, the regularization procedure improves the results obtained with graph clustering.

CHAPTER 5

Wave Kernel Embedding

5.1 Introduction

In pattern recognition, Graph embeddings have found widespread use for the purposes of clustering, analyzing and visualizing relational data. However, they have also proved to be useful as a means of graph characterization. There are many examples in the literature including ISOmap (Tenenbaum et al., 2000), the Laplacian eigenmap (Belkin & Niyogi, 2002), and the heat-kernel embedding (Xiao, Hancock & HangYu, 2010), to name a few. Once embedded, a graph can be characterized using a feature-vector that characterizes the point-set distribution resulting from the embedding (Xiao, Hancock & Wilson, 2009). This kind of representation is convenient since a Euclidean vector space makes available powerful geometric analysis tools for data analysis, not available for discrete or structural representations. However, such an embedding assumes that the original relational data is metric. Sometimes, however, this is not the case. This is the case when the matrix characterization of the relational graph contains negative eigenvalues, i.e. it is not positive semi-definite. Under these circumstances, the

graph embeds not into a Euclidean space, but into pseudo-Euclidean or Krein space (Pekalska & Haasdonk, 2009). This problem has attracted relatively little attention in the literature. Our aim in this chapter is to embed the nodes of a graph as points on the surface of a pseudo-Riemannian manifold in a pseudo-Euclidean space, and to use the resulting point-set as the basis from which to compute graph characteristics. To provide a framework for our study, we turn to the wave kernel. This is the solution of a wave equation, which is an important second-order linear partial differential equation that describes the propagation of a variety of waves. Crucially, the solutions are complex and therefore reside in a pseudo-Euclidean space. Although the wave equation has been extensively studied in the continuous domain, there has been relatively little effort devoted to understanding its behavior on a graph. In common with the heat kernel, the wave kernel can be defined in terms of a combinatorial Laplacian. However, in the case of the wave kernel this is the edge-based Laplacian, introduced by Friedman (Friedman & Tillich, 2004b).

In this chapter we explore how to solve the edge-based wave equation, in terms of the eigensystem of the edge-based Laplacian. Since the solution is a sinusoid, it contains both real and imaginary parts. Hence, we embed the nodes of the graph as points residing on a pseudo-Riemannian manifold, determined by the eigenvalues and eigenvectors of the edge-based Laplacian. In our experiments on graphs extracted from 2D image data, we use this matrix for the purpose of graph matching. The remainder of this chapter is organized as follows: In Section5.2 we commence by embedding graphs onto pseudo Riemannian manifolds. First we show how to find the solution of the wave equation on a graph using the edge-based Laplacian in Section5.2.1. Then we construct the coordinate matrix for the pseudo-Euclidean embedding in Section5.2.3. Finally, Section5.2.2 is devoted to establishing the edge-based Laplacian matrix. In Section 5.3 we illustrate how to manipulate vectors in a pseudo Euclidean space, commencing by computing the square distance between any arbitrary pair of vectors in Section5.3.1. In Section5.3.2 we show how to construct an orthonormal basis, and in Section5.3.3 how to project vectors from a pseudo Euclidean space

onto a 2D sub-space. Section 5.4 presents our experimental evaluaton. Finally, a couple of conclusions are drawn in Section 5.5.

5.2 Embedding graphs into Pseudo Riemannian manifolds

This section is dedicated for the mathematical foundations needed for constructing the wave kernel to embed the graphs into pseudo Riemannian manifolds. Commencing by solving the so-called Edge-Based Wave Equation in Section 5.2.1, followed by constructing the matrix whose columns are the coordinates of the nodes residing on the pseudo Euclidean space. Section 5.2.2 is dedicated for introducing the concept of the edge-based Laplacian matrix.

5.2.1 Edge-based Wave Equation

Recently, Friedman in (Friedman & Tillich, 2004b) has developed a graph-based version of the wave equation that has many of the properties of the classical Laplacian wave equation. The development is based on a variant of the combinatorial Laplacian referred to as the edge-based Laplacian. This graph theoretic version of the wave equation provides an interesting link with the continuous wave eqaution, and has a simple physical interpretation. The edges of the graph can be viewed as taut strings, joined together at the vertices. In fact, the edge-based Laplacian has been shown in the physics literature to be the "limiting case" of a "quantum wire" (Hurt, 2000).

Classical Graph theory defines a combinatorial Laplacian, L, as an operator on functions whose domain is the set of vertices of a graph. On the vertex-set the wave equation is $U_{tt} = -LU$ (the negative sign is due to that the combinatorial Laplacians are positive semi-definite). However, this wave equation fails to give a finite speed for wave propagation. As a result there is no simple link between the graph theoretic wave equation and its continuous counterpart. To overcome this problem a so-called edge-based wave equation $W_{tt} = -L_E W$

was introdiced by Friedman (Friedman & Tillich, 2004b), where L_E is the edge-based Laplacian, which is a better approximation to the continuous Laplacian (i.e. the second derivative) than the combinatorial Laplacian L. The edge-based wave equation has unit wave propagation speed, while that based on the combinatorial Laplacian L has infinite speed.

For the edge-based Laplacian, the eigenfunction f satisfies $L_E f = \lambda f$ and $Lf = 0$ where $\Lambda_E = \{\lambda\}$ is the set of Laplacian eigenvalues. In fact, if L is normalized and the graph under study has each edge weight equal to unity, then L is similar to $\left(1 - \cos\sqrt{L_E}\right)$. That is to say if Δ is a continuous Laplacian then $\widetilde{\Delta} = 1 - \cos\sqrt{-\Delta}$ is the corresponding combinatorial Laplacian. Hence, the eigenvalue λ is in Λ_E if and only if $\left(1 - \cos\sqrt{\lambda}\right)$ is in Λ (the set of all eigenvalues of the combinatorial Laplacians). Note that Λ_E is an infinite set of non-negative values (whose square roots are periodic), and exclude those which are multiples of π from Λ_E. The general solution of the wave equation (Folland, 1995)

$$\begin{aligned} W_{tt} &= -L_E W \\ W|_{t=0} &= f \\ W_t|_{t=0} &= g \end{aligned} \tag{5.1}$$

has the form

$$W = \frac{\sin\left(\sqrt{L_E}t\right)}{\sqrt{L_E}} g + \cos\left(\sqrt{L_E}t\right) f \tag{5.2}$$

For our work it suffices to compute the fundamental solution W that satisfies $W|_{t=0} = 0$ and $W_t|_{t=0} = 1$, that is

$$W = \frac{\sin\sqrt{L_E}t}{\sqrt{L_E}} \tag{5.3}$$

Since, L_E is positive semi-definite (Friedman & Tillich, 2004b), W can be ap-

proximated using the MacLaurin series, giving

$$W = t[I - \frac{1}{6}L_E t^2 + ...] \tag{5.4}$$

Now we can consider the nodes of the graph as residing on a pseudo-Riemannian manifold and the edges as geodesics on the manifold.

5.2.2 Edge-based Eigenvalues and Eigenfunctions

In the previous section we presented the methodology for constructing the wave kernel embedding matrix. To commence, we need first to construct the edge-based Laplacian matrix. We follow the procedure given in (Friedman & Tillich, 2004b) where the edge-based eigenvalues and eigenfunctions are determined using those of a normalized adjacency matrix. To commence, consider a finite graph denoted by $G = (V, E)$ with node-set V and edge-set of edges $E \subseteq V \times V$, with all edges of unit weight. The elements of the adjacency matrix A of the graph G are

$$A(u, v) = \begin{cases} 1 & if (u, v) \in E \\ 0 & otherwise \end{cases} \tag{5.5}$$

Let T be a diagonal matrix whose elements are the degrees of the nodes of G, that is $T(u, u) = \sum_{v \in V} A(u, v) = \textit{deg}_u$. By dividing each row of the adjacency matrix A by its corresponding $\textit{deg}_u$, we obtain the normalized adjacency matrix $\widetilde{A}$. For each eigenvalue, λ of $\widetilde{A}$ there is a unique value of $\cos^{-1}(\lambda) \in [0, \pi]$. The edge-based eigenvalues are $2n\pi + \cos^{-1}(\lambda)$ and $2(n + 1)\pi - \cos^{-1}(\lambda)$, where $\{n = 0, 1, 2, ...\}$. Hence, if $\omega \in \{\Re \setminus n\pi\}$ then ω^2 is an edge-based eigenvalue if and only if $\cos \omega$ is an eigenvalue of $\widetilde{A}$. For each corresponding eigenfunction, *f*, of $\widetilde{A}$, *f* can be extended to obtain an edge-based eigenfunction (Friedman & Tillich, 2004b). To summarize, for the edge-based eigenpair (f, λ), we have that:

1- $\cos \lambda$ is an eigenvalue of $\widetilde{A}$,

2- *f* is an eigenfunction of $\widetilde{A}$; that is $\widetilde{A}f = \cos \lambda\ f$,

3- $L_E f = \lambda f$ and $Lf = 0$.

The existence of a complete set of eigenvalues and eigenfunctions for the contin-

uous Laplacian has been demonstrated in (Gilbarg & Trudinger., 1983). Friedman has extended the analysis to the edge-based Laplacian for finite graphs (Friedman & Tillich, 2004b). To outline the theory, let G be a finite graph. For G there exists eigenpairs f_i, λ_i for the edge-based Lalacian, such that

1- $0 \leqq \lambda_1 \leqq \lambda_2 \leqq ...$,

2- f_i satisfies the Dirichlet (Neumann) conditions[1] (Arendt & Warma, 2003),

3- f_i forms a complete orthonormal basis for $L^2_{Dir}(G)$ $(L^2(G))$ (Arendt & Warma, 2003),

4- $\lambda_i \to \infty$.

Physically, the equations $L_E f = \lambda f$ and $Lf = 0$ describe the vibrational modes associated with a taut strings on each edge that are joined together at the vertices. If we excite or "pluck" the system, it would produce tones with frequencies $\sqrt{\lambda}$, with λ ranging over the edge-based eigenvalues. That is to say, the spectrum of the edge-based Laplacian gives the number of harmonics corresponding to the harmonic frequencies of vibrations of the edges of the graph. In a more computational sense, it would be like as we projected the nodes of the graph into an eigenspace spanned by the harmonic frequencies corresponds to the eigenfunctions of the Laplacian.

5.2.3 The manifold spanned by the data

Positive definite Riemannian manifolds can be represented in one of two ways. Either a) their properties are defined intrinsically, or b) they can be regarded as subsets of a Euclidean space of higher dimension. Following the work of Nash (Nash, 1954, 1956) and Whitney (Whitney, 1936), it has been known for some time that these approaches are equivalent, in the sense that any intrinsically defined Riemannian manifold can be embedded, with appropriate differentiability, into a Euclidean space. In (Clarke, 1970), Clarke showed that the same situation holds in the case of pseudo-Riemannian manifolds, with metrics of indefinite signature.

[1]The Dirichlet (Neumann) boundary conditions specify the value (the normal derivative) of the function on a surface.

The pseudo-Euclidean space generalizes the well-known Euclidean space to the case where inner products are indefinite. This effectively amounts to two Euclidean spaces, one of which has a positive semi-definite inner product and the second with a negative semi-definite inner product. For the kernel matrix, the embedding is determined by the pseudo Gram matrix $C = -\frac{1}{2}QWQ$ derived from the kernel matrix W, where $Q = I - \frac{1}{n}ee^T$and $e = (1, ..., 1)^T$. If the matrix with the embedding co-ordinates as columns is X, then

$$C = X^T X \tag{5.6}$$

In the non-Euclidean case where W and C are not positive semi-definite, a method to define a pseudo-Euclidean space was given in (?, ?) such that

$$C = X^T \begin{pmatrix} M & 0_{p+q\times k} \\ 0_{k\times p+q} & 0_{k\times k} \end{pmatrix} X \tag{5.7}$$

with

$$M = \begin{pmatrix} I_{p\times p} & 0_{p\times q} \\ 0_{q\times p} & -I_{q\times q} \end{pmatrix} \tag{5.8}$$

$0_{k\times k}$ is the $k \times k$ matrix filled with zeros, p and q are the numbers of positive and negative eigenvalues of C respectively and $p + q + k = n$. We can then write $X^T MX = \Phi\Lambda\Phi^T = \Phi|\Lambda|^{\frac{1}{2}}M|\Lambda|^{\frac{1}{2}}\Phi^T$, where Φ is the column-matrix of the eigenvectors and Λ the diagonal matrix of the corresponding eigenvalues. The vectors are recovered via the transformation $X_L = |\Lambda_L|^{\frac{1}{2}}\Phi_L^T$, where Φ_L is the column-matrix of the selected eigenvectors and Λ_L the diagonal matrix of the corresponding eigenvalues. Hence, the columns of X_L are the vectors in the pseudo-Euclidean space.

5.3 Pseudo Euclidean Space

A pseudo Euclidean space is an n-d imensional space $r_1, r_2, ..., r_n$ where $r_i = r$ or ir and r is a set of real numbers and $i = \sqrt{-1}$. In this section we describe how

to manipulate vectors in a pseudo Euclidean space. Firstly, we explain how to compute the square distance between any arbitrary pair of vectors. Secondly, we show how to construct an orthonormal basis. Thirdly, we show how to project vectors from a pseudo Euclidean space onto a 2D sub-space.

5.3.1 Distance Function

With a pseudo Euclidean space R^n we assign a symmetric bilinear form $\rho : R^n \times R^n \rightarrow R$, $\rho(x,y) = x^T S y$ where S is the matrix whose elements $s_{ij} = \frac{1}{2}(d_{i0}^2 + d_{j0}^2 - d_{ij}^2)$; d is a distance function with pairwise distances d_{ij} for all $1 \leqslant i,j \leqslant n$. For any two vectors $x, y \in R^n$, $\rho(x,y)$ is the inner product of x and y and $\|x-y\|^2 = \rho(x-y, x-y)$ is the squared distance between x and y. Since S is real symmetric, there is an orthogonal matrix Ψ and a diagonal matrix Γ such that $S = \Psi\Gamma\Psi^T$, the elements δ_i of Γ are the eigenvalues of S arranged in order and the columns of Ψ are the correspondingly ordered eigenvectors. It is worth mentionng that if the matrix S has negative eigenvalues, then the squared distance between two vectors in the pseudo Euclidean space may be negative. It for this reason that we do not speak about "distance" between vectors in pseudo Euclidean space. Moreover, the fact that the squared distance between two vectors vanishes does not imply that these two vectors are the same. This is not the case in a Euclidean space.

5.3.2 An Orthonormal Basis

The columns $\{b_i\}, i = 1, ..., n$ of the matrix $B = I\Psi$ represent an orthogonal basis of R^n, since S is the matrix of ρ with respect to the natural basis $\{e_i\}, i = 1, ..., n$ where $e_i = (0, ..., 1_i, ..., 0)$. We can therefore write the bilinear form ρ with respect to the basis b_i as $S_b = \Psi^T S \Psi$, so that $S_b = \Gamma$. For any two vectors x and y in R^n, $\rho(x,y) = x^T S y = [x_b^T \Psi^T][\Psi S_b \Psi^T][\Psi y_b \Psi^T]$. Hence, $\rho(x,y) = x_b^T S_b y_b = x_b^T \Gamma y_b$. Accordingly, the inner product of x and y can be written as $\rho(x,y) = \Sigma_{i=1}^n \delta_i (x_b)_i (y_b)_i$ and the squared distance as $\|x-y\|^2 = \Sigma_{i=1}^n \delta_i \left([x_b]_i - [y_b]_i\right)^2$. The matrix $X_b = X\Psi$ has as columns the coordinates

with respect to the basis $\{b_i\}$. Conversely, the coordinate matrix $X_e = X_o\Psi^T$ has as columns the co-ordinate vectors with respect to an orthogonal natural basis.

Let us define a diagonal matrix $J = diag(j_{ij})$ with elements

$$j_i = \begin{cases} 1 & \delta_i > 0 \\ 0 & \delta_i = 0 \\ -1 & \delta_i < 0 \end{cases}$$

and $i = 1, \ldots, n$, Furthermore, let $\widetilde{\Gamma} = diag(\gamma_i)$ where

$$\gamma_i = \begin{cases} |\delta_i| & if\ \delta_i \neq 0 \\ 1 & otherwise \end{cases}$$

Now consider the matrix $\widetilde{\Psi} = \Psi\widetilde{\Gamma}^{-\frac{1}{2}}$. The first l columns of this matrix are orthonormal vectors with respect to $\{b_i\}$. To show this consider the matrix

$$\widetilde{\Psi}^T S\widetilde{\Psi} = \widetilde{\Gamma}^{-\frac{1}{2}}\Psi^T[\Psi S_b\Psi^T]\Psi\widetilde{\Gamma}^{-\frac{1}{2}} = \widetilde{\Gamma}^{-\frac{1}{2}}\Gamma\widetilde{\Gamma}^{-\frac{1}{2}} = J$$

The diagonal elements $J_i, i = 1, \ldots, l$ are either 1 or -1, while the remainder are zeros. Hence, the first l columns of the matrix $\widetilde{B} = B\widetilde{\Gamma}^{-\frac{1}{2}}$ form an orthonormal basis of R^l. Finally, for the matrix X_e whose columns are the co-ordinate vectors in the pseudo Euclidean space with respect to the natural basis $\{e_i\}_{i=1,\ldots,n}$, the corresponding matrix of coordinates with respect to the orthonormal basis $\{b_i\}_{i=1,\ldots,n}$ is $X_{\widetilde{b}} = X_e\widetilde{\Psi}$.

5.3.3 Projection into a kD Subspace

Suppose we order eigenvalues of the matrix S so that first l^+ eigenvalues are positive, the following l^- are negative and the remaining are zeros, where $l = l^+ + l^-$. As a result $\{b_i\}_{1\leq i\leq l}$, and the first l columns of the matrix B given in Section 5.3.2 form an orthogonal basis of the space R^l. Using the first l columns of the matrix $\widetilde{\Psi}$, we can locate the projections of the column vectors of the ma-

trix X onto the space R^l with respect to $\{b_i\}_{1\leq i\leq l}$ as $X_l = B\widetilde{\Psi}^T$. To obtain the coordinates of X_l with respect to the orthonormal basis $\widetilde{b_i}$, we construct the matrix $X_{l_{\tilde{b}}} = \widetilde{B}\widetilde{D}_l^{-\frac{1}{2}}\Psi_l^T = (\overline{p}_1|\overline{p}_2|...)$, where $\widetilde{D}_l = diag(\gamma_i), 1 \leq i \leq l$ is the l^{th} leading principle submatrix of $\widetilde{D}$ and $\overline{p}_i$ is the projected coordinate vector of the i^{th} node of G. Again we can define the inner product of two arbitrary vectors, $x\ and\ y$, as $\rho(x,y) = \Sigma_{i=1}^n \delta_i (x_b)_i (y_b)_i$ and the squared distance as $\|x-y\|^2 = \Sigma_{i=1}^n \delta_i \left([x_b]_i - [y_b]_i\right)^2$.

To avoid problems associated with dealing with a space of high dimensionality, we will ignore the dimensions for which the eigenvalues are small in magnitude. Therefore, if we arrange the eigenvalues in descending order by their absolute values, the first k eigenvalues (typically $k = 2\ or\ 3$) where $k < l$ will span a space R^k in which we can project the exact vector representation of the pseudo Euclidean space R^n.

5.4 Experiments and Results

We experiment with the wave kernel embedding as a graph characterization for the purposes of graph-matching. We represent the graphs under study using sets of coordinate vectors corresponding to the embedded node position, and compute the similarity of the sets resulting from different graphs using the robust modified Hausdorff distance.

In our experiments we aim to investigate whether the edge-based wave kernel embedding can be used as a graph characterization, for gauging the similarity of graphs, and hence clustering them. To commence, we compute the eigensystem of the edge-based Laplacian from the eigensystem of the normalized adjacency matrix, and hence compute the edge-based Laplacian matrix introduced in Section 5.2.2. The edge-based wave kernel then is computed as described in Section 5.2.1 with the values of $t = 10.0, 1.0, 0.1$ and 0.01. From the wave-kernel we compute the embedding coordinate matrix, whose columns are the coordinates of the embedded nodes in a pseudo-Euclidean space. Finally, we project the co-ordinate vectors onto a pseudo-Euclidean space with low dimension using

	t=10	t=1.0	t=0.1	t=0.01
Houses data	0.2333	0.0000	0.0333	0.1000
COIL data	0.3333	0.3333	0.3333	0.7000

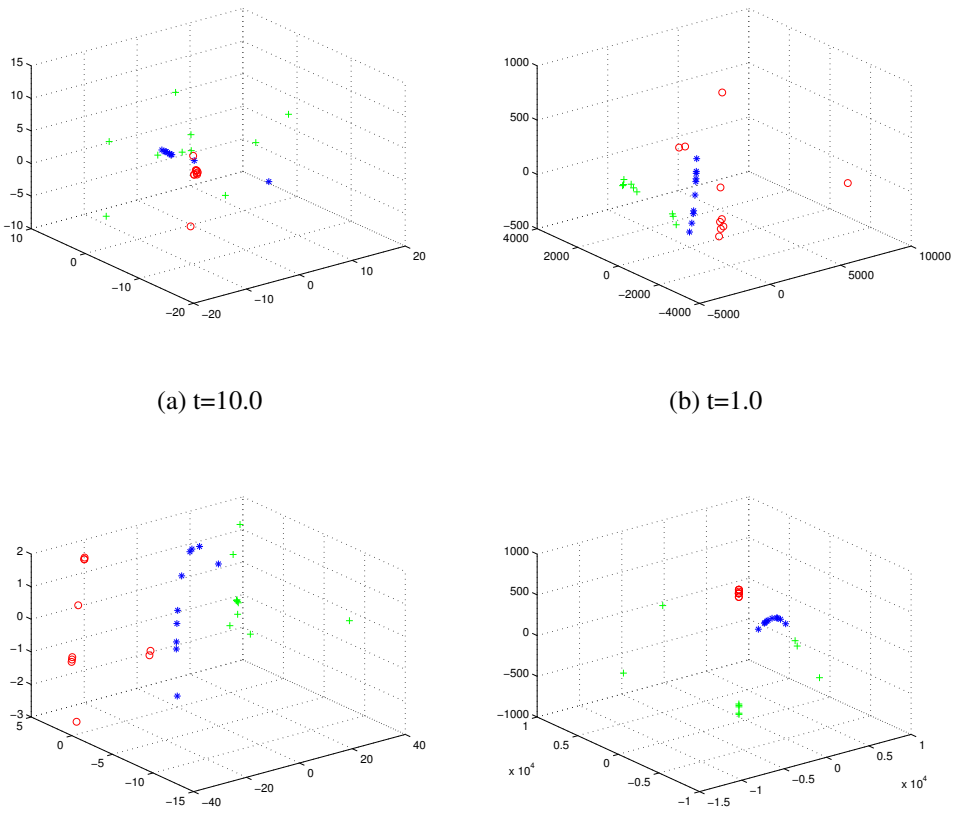
Table 5.1: A rand index vs. t for the York model house dataset

the orthonormal basis as shown in Section 5.3. With the vector representations residing in a low dimension space we construct the distance matrices between the thirty different graphs using both the classical and modified Hausdorff distance 3.4.1. Finally, we subject the distance matrices to multidimensional scaling MDS (Cox & Cox, 1994) to embed them into a 3D space. Here each graph is represented by a single point. Figure 5.2 shows the results obtained using the modified Hausdorff distance. The subfigures are ordered from left to right (up to down), using $t = 10.0, 1.0, 0.1$ and 0.01 in the wave kernel. We have also investigated the COIL data, and the results are shown in Figure 5.4.

We commence by introducing the results obtained when experimenting with the York model house database. Where the CMU model house sequence is represented as a red circle and each graph of the MOVI model house sequence is represented as blue star while each graph of the Swiss chalet model house sequence is represented as a green cross. To commence, we show in Figures 5.1 and 5.2 the results when using the Hausdorff distance (HD) and the modified Hausdorff distance (MHD) to measure the (dis)similarity between pairs of graphs, respectively. The subfigures are ordered from left to right, top to bottom using the heat kernel embedding with the values $t = 10.0, 1.0, 0.1$ and 0.01 respectively. With the same order, Figures 5.3 and 5.4, give the results obtained when using COIL-20 dataset where each graph of the sequence of the first object is represented as a red circle and each graph of the sequence of the second object is represented as blue star while each graph of the sequence of the third object is represented as a green cross..

To investigate the results in more details table 4.1 shows the rand index for the distance as a function of t. This index is computed as explained in Section 3.4.5.

Although, the wave kernel gives a reasonable separation of the objects into



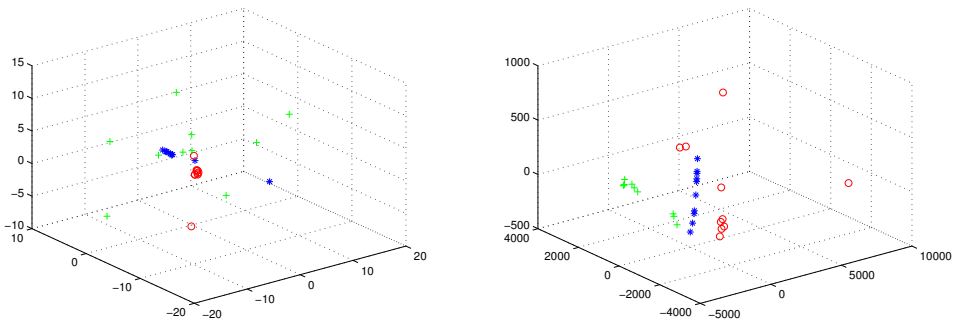
(a) t=10.0

(b) t=1.0

(c) t=0.1

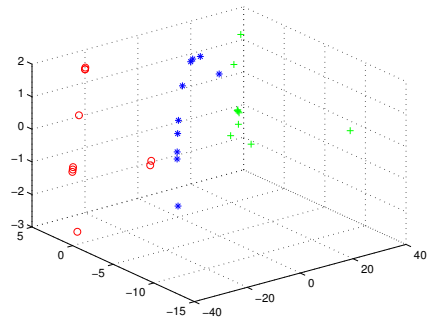
(d) t=0.01

Figure 5.1: MDS embedding obtained when using HD for the Wave Kernel embedding for the houses data.

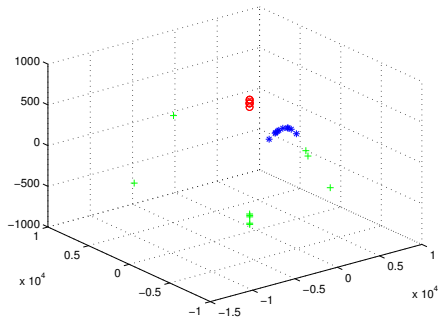


(a) t=10.0

(b) t=1.0

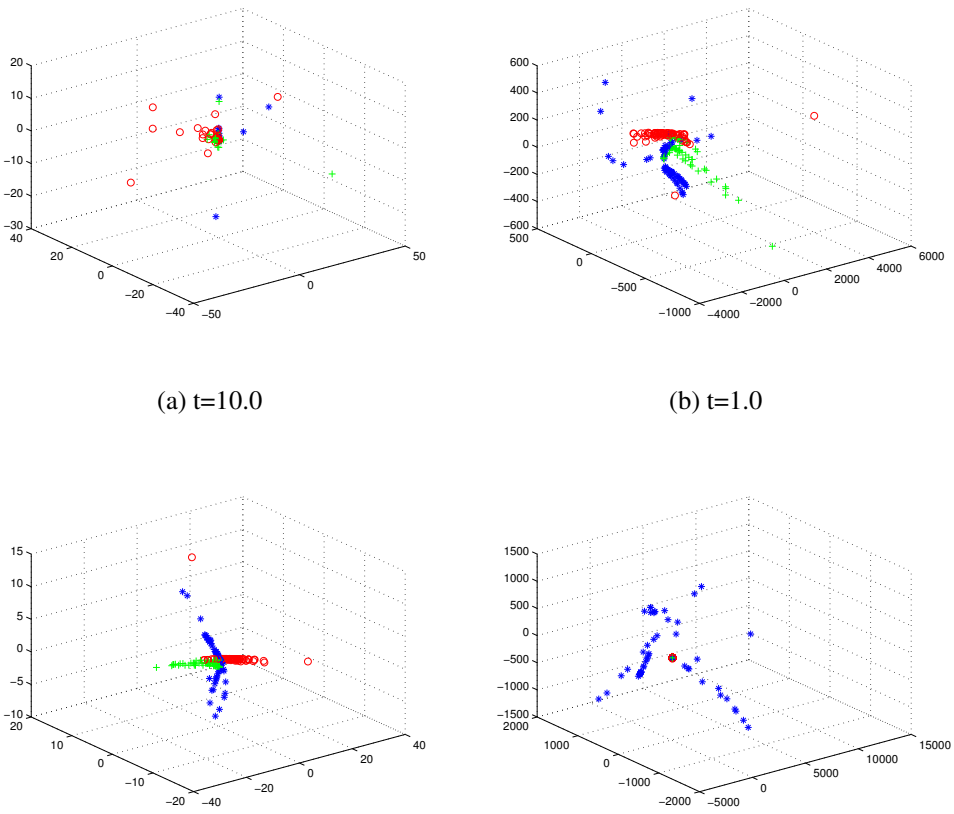


(c) t=0.1



(d) t=0.01

Figure 5.2: MDS embedding obtained when using MHD for the Wave Kernel embedding for the houses data.



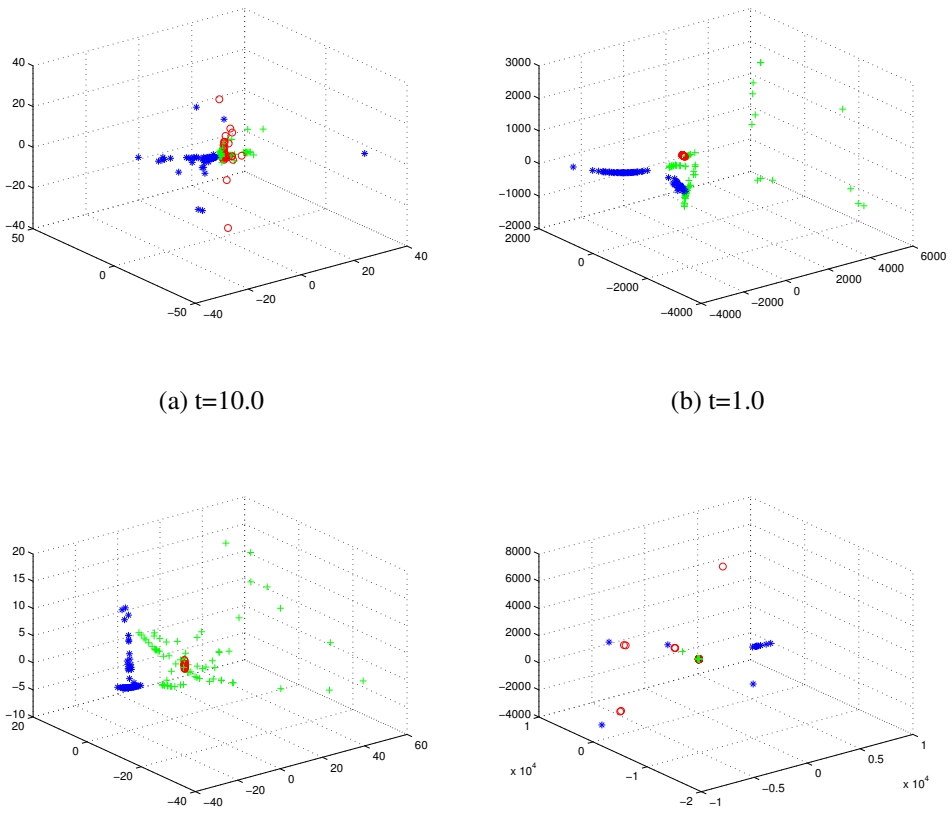
(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 5.3: MDS embedding obtained when using HD for the Wave Kernel embedding for the COIL data .



(a) t=10.0

(b) t=1.0

(c) t=0.1

(d) t=0.01

Figure 5.4: MDS embedding obtained when using MHD for the Wave Kernel embedding for the COIL data .

distinct clusters particularly for value of t equal to 1, the experimental study shows that it needs further development; for that reason is we suggest in Appendix I to generalize the wave kernel framework in higher dimensional space. However, we can figure out a number of conclusions to be drawn from the plots. For instance, the sequence of the second object of the COIL dataset is clustered along a straight line for all values of t, while the other two sequence are embedded in a less compact cluster. Whilst, the York model house dataset gives a more obvious clusters than those of the COIL dataset. Unlike the situation when using the heat kernel embedding where we project the data into a positive space, in the wave kernel case we try to preserve the geometry of the original data.

5.5 Conclusion

In this chapter we have established a procedure to embed the nodes of a graph into a pseudo-Riemannian manifold based on the wave kernel, which is the solution of an edge-based wave equation. Under the embedding, each edge became a geodesic on the manifold. The eigensystem of the wave-kernel was determined by the eigenvalues and the eigenfunctions of the normalized adjacency matrix. By factorizing the Gram-matrix for the wave-kernel, we determine the embedding co-ordinates for nodes under the wave-kernel. We investigated the utility of this new embedding as a means of gauging the similarity of graphs. We experimented on sets of graphs representing the proximity of image features in different views of different objects from two datasets (the York Model House and COIL datasets). And by applying multidimensional scaling to the similarity matrix we demonstrated that the proposed graph representation is capable of clustering different views of the same object together.

CHAPTER 6

Conclusion and Future Work

In this chapter we commence by a summary of the main contributions of the thesis along with the conclusions driven. This includes the novel ideas on the geometric graph characterisation, p-Laplacian graph regularisation framework and the graph wave kernel embedding. Then we draw on some of the drawbacks and possible extensions of the work.

6.1 Summary and Conclusion

The overall objective of this thesis was to develop a framework for graph characterisation by combining methods from both spectral graph theory and manifold learning theory, and to explore whether they can provide a stable and robust graph representation. When we began to study these methods, we suggested a few sub-goals to be achieved while heading for the main goal, for instance we were aiming to extract stable and robust geometric invariants that can be used for characterizing the graphs aiming at preserving the local manifold structure and to represent the graph by a set of curvatures associated with its edges, nodes or

triangular faces.

For doing so, in Chapter 3 we used two powerful mathematical tools, namely the spectral graph theory and the heat-diffusion in Riemannian geometry, to analyse the data from the perspective of the intrinsic geometry. We commenced by using the heat kernel to provide a geometric characterization of graphs by the means of the spectral geometry of the combinatorial Laplacian. Performing the Young-Householder decomposition to the heat-kernel maps the nodes of the graph to points in the manifold, the decomposition provided a matrix of the embedding point position vector. The embedding offers the advantage that time parameter can be used to control the condensation of the clusters. With the embeddings to hand, we developed a graph characterization based on differential geometry using the notation of the sectional curvatures associated with the edges of the graph, making use of the fact that the sectional curvature can be determined by the difference between the geodesic and Euclidean distances between pairs of nodes. Furthermore, we used the Gauss-Bonnet theorem to compute the Gaussian curvatures associated with triangular faces of the graph as. With sets of curvatures, defined either on the edges or the faces of the graph under study we constructed the graph characterisation matrices, to be used for the purpose of graph matching and clustering. To this end, we computed the similarities of the graphs using robust variants of the Hausdorff distance which allows us to compute the similarity of different graphs without knowing the correspondences between graph edges or faces. In the approach we proposed, we kept in mind the same concept as the classical manifold learning techniques; where the manifold is considered to be embedded linearly or almost linearly in the ambient space. Nevertheless the main target was to preserve the local affine geometric structures in the neighborhoods around each data point. While, classical techniques regard more global features.

Due to the noise and the inaccurate estimation in real applications, we aimed next to use a manifold regularization to overcome the effects of noise while preserving the geometrical structures existing in the initial data. For that reason, in Chapter 4 we proposed a framework to regularize real-valued or vector-valued

functions on weighted graphs of arbitrary topology based on the p-Laplacian operators . The proposed framework brought together several distinct concepts that have received some independent recent attention in machine learning. The first of these was the methods deduced from the spectral graph theory which have been widely used for clustering and classification tasks over the last decades. Even more, we considered the properties of the graph p-Laplacian as a nonlinear extension of the usual graph Laplacian. The second concept to be used was the geometry coming from the heat kernel embedding of the graph into a manifold in a class of algorithms known as manifold learning. In these techniques we used the geometry of the manifold by assuming that it has the geometric structure of a Riemannian manifold. The third important conceptual framework came from the Manifold regularization, which extends the classical framework of regularization in the sense of reproducing Hilbert Spaces to exploit the geometry of the embedded set of points. In the proposed approach, we have considered minimizing a weighted sum of two energy terms. The first one is a regularization term that uses the discrete p-Dirichlet form; with the degree ,p, of regularity. The second one is an additional approximation term which helps to avoid the shrinkage effects obtained during the regularization process with appropriate choice for the regularization parameter λ. The existing regularization techniques use only the regularization term which may lead the curves on the manifold to shrink and hence become a point.

When the initial data lie in a high-dimensional space, one great challenge is to be able to map the data into a lower dimensional space such that standard methods could be efficiently applied. Moreover, in many cases the data lie on a nonlinear manifold, but neither the actual structure nor the dimension of the latter is known in advance. To this point, we come to the last contribution in this thesis; that is to construct the Wave kernel embedding matrix which is mainly based on the edge based Laplacian. The concept comes from a recently developed graph calculus. The Calculus enables more analysis techniques to carry over to graphs and vice versa in a very simple way; that is less intuition is obscured in technicalities that are particular to analysis or graphs. In particular, most techniques

for the non-linear p-Laplacian in analysis to be easily carried over to graph theory. This allows the use of the non-linear functions; those functions which are not edgewise linear. In Chapter 5, we've described a new framework for embedding graphs on pseudo-Riemannian manifolds based on the wave kernel; which is the solution of the wave equation on the edges of a graph. The eigensystem of the wave-kernel is determined by the eigenvalues and the eigenfunctions of the normalized adjacency matrix and can be used to solve the edge-based wave equation. By factorising the Gram-matrix for the wave-kernel, we determine the embedding co-ordinates for nodes under the wave-kernel. We investigate the utility of this new embedding as a means of gauging the similarity of graphs.

6.2 Future Work

In summary, we have introduced a novel framework for pattern and object recognition with interesting mathematical and computational properties. However, this framework needs to be experimented more with different types of data, such as characters, fingerprints, documents, and images. There are a number of ways in which the work reported in this thesis can be extended. For instance, to continue the work done in Chapter 3, it would be interesting to explore the use of the curvatures as a means of directly embedding the nodes of the graphs on a manifold. And it would be interesting to investigate if the curvatures could be used to aid the process of visualising or drawing graphs. As a continuation of the work done in Chapter 4, we plan to extend this regularization framework to other fields and to combine it to others techniques However, there are still several questions remaining to be investigated in our future work; for instance, it remains unclear how the regularization parameter λ, controls the smoothness in our approach, theoretically and efficiently. Actually, the regularization framework is also interesting to help to estimate some geometric and topological features such as normals, curvatures, or shape skeletons. Moreover, discovering the structure of the manifold from a set of data points sampled from the manifold with noise is still a very challenging concept.

In Chapter 5, we began a work based on some techniques of the graph calculus partial differential functions to problem of graph embedding, matching and clustering. We're interested in applying more ideas and techniques from the graph calculus of partial differential functions to the purposes graph matching and clustering. More generally, we are interested in how the philosophy of calculus can be used to tackle problems in other applications and fields. At the end there are many questions to be asked: Is the assumption of the manifold structure sensible? How to make the algorithm more robust to noises? Can we develop a framework to discover the underlying low dimensional manifold structure?

For future research we'll turn our attention to a recently discovered branch of differential geometry, known as "Generalized Geometry", which has received a reasonable amount of interest inspired by its connections with areas of Mathematical Physics. The theory is also of interest because the different geometrical structures are often generalizations of more familiar geometries. In this sense, we aim to generalize the framework we introduced in Chapter 5 in higher dimensional space. This generalization framework may lead us to use space-time geometry for more application in physics.

Part I

Appendices

APPENDIX A

Area Metric Manifolds

A.1 Area Metric Geometry

Generalized geometries commence to play a progressively more significant role, in spite of the fact that one initial starting point for its formulation is a metric target manifold. The emerging picture is that area metric manifolds are generalized geometries. An area metric may be defined as a fourth rank tensor field which allows to assign a measure to two-dimensional tangent areas, in close analogy to the way a metric assigns a measure to tangent vectors. In more than three dimensions, area metric geometry is a true generalization of metric geometry; although every metric induces an area metric, not every area metric comes from an underlying metric. The essential features of area metric geometry, to the extent that they are of relevance to this chapter, are presented and discussed in this section.

A.1.1 Area Metric Manifolds

An area metric manifold (M,G) is a smooth differential manifold equipped with an algebraic curvature map G, which is a smooth covariant fourth rank tensor

field $G : (T_pM)^{\otimes 4} \to \Re$ satisfying the following symmetry (i, ii) and cyclicity (iii) properties at each point of the manifold:

$$\begin{aligned} &For\, all\, vector\, fields\, X, Y, Z, A\, and\, B\, in\, TM \\ &(i)\ G_{XYAB} = G_{ABXY}, \\ &(ii)\ G_{XYAB} = -G_{YXAB}, \\ &(iii)\ G_{AXYZ} + G_{AYZX} + G_{AZXY} = 0. \end{aligned} \tag{A.1}$$

By the first two conditions (i, ii), an algebraic curvature map has two index pairs which can be symmetrically exchanged. Hence G naturally provides us with a linear map from the space of antisymmetric contravariant two-tensors $\Lambda^2 TM$ to its dual, i.e., a map $G : \Lambda^2 T_pM \otimes \Lambda^2 T_pM \to \Re$. Furthermore, G has an inverse corresponding to a map:

$$\begin{aligned} &G^{-1} : \Lambda^2 T^*M \otimes \Lambda^2 T^*M \to \Re,\ such\, as \\ &G^{-1}G = id_{\Lambda^2 T_pM} \end{aligned} \tag{A.2}$$

Where $\Lambda^2 T_pM$ denotes the space of all contravariant antisymmetric tensors of rank two.

It's of significant to notice that in three dimensions every area metric is metric-induced; from four dimensions onwards, however, there exist area metrics that cannot be induced from any metric. Nevertheless, it is interesting to discuss the following special type of area metrics:

A.1.2 Induced-metric area metric

A special case of area metric manifolds is metric manifolds (M, g), since any metric manifold is an area metric manifold (M, Gg), by virtue of

$$Gg_{XYAB} = g_{XA}g_{YB} - g_{XB}g_{YA}. \tag{A.3}$$

for which expression the properties of an algebraic curvature map are readily checked. This construction has a clear geometrical interpretation. Knowing how

to measure lengths and angles, one can measure areas: the expression Gg_{XYXY} returns precisely the squared area of the parallelogram spanned by the vectors X and Y . The converse, however, does not hold: the ability to measure areas does not imply a length measure. Hence an area metric is a weaker structure than a metric, as length measurement implies area measurement but not vice versa. Importantly, while the definition of a general area metric manifold keeps all algebraic properties of the metric-induced case, an area metric admits more degrees of freedom than a standard metric. This fact is related to the decomposition theorem recently introduced by Gilkey, which states that any algebraic curvature map G can be written as a linear combination of algebraic curvature maps that are induced from a finite collection of metrics $g^i|i = 1...N$, in the form

$$\begin{aligned} G &= \sum_{i=1}^{N} \sigma^i G g^i, \\ \sigma^i &= \pm 1. \end{aligned} \tag{A.4}$$

Unfortunately, this decomposition is far from unique. Moreover, no constructive algorithm for the decomposition of an arbitrary algebraic curvature map into metric-induced maps is currently known [19]. Correspondingly, it is an open question how many metrics are required for the decomposition of a given algebraic curvature map, although it is known that the number of required metrics, in $d-$ dimensions, is certainly bounded from above by $d(d+1)/2)$ [20].

A.2 The space of oriented areas

In the geometry of area metric manifolds, areas play a role analogous to the one of vectors in the geometry of metric manifolds. However, the space of areas over a tangent space T_pM at any point p is not a linear space. To be more precise, we observe that on the vector space of parallelograms $\Lambda^2 T_pM \otimes \Lambda^2 T_pM$, each parallelogram $X \wedge Y$ is spanned by two vectors (X and Y) in the same tangent space(including the degenerate case where these vectors are linearly dependent). Clearly, $X \wedge Y$ is an element of the vector space $\Lambda^2 T_pM$, but a generic antisym-

metric two-tensor $\Omega \in \Lambda^2 T_p M$ may only be written as the product of two vectors if $\Omega\Lambda\Omega = 0$. Such elements of $\Lambda^2 T_p M$ are called simple and constitute the space of oriented areas $A^2 T_p M$. This finally yields the identification of the oriented areas as a polynomial subset of the vector space of antisymmetric two-tensors:

$$A^2 T_p M = \{\Omega \in \Lambda^2 T_P M | \Omega\Lambda\Omega = 0\} \tag{A.5}$$

Polynomial subsets of vector spaces are called varieties in algebraic geometry [23], and so the area space is a variety embedded in $\Lambda^2 T_P M$. Strictly speaking, the area metric G only ought to act on the variety $A^2 T_p M$, which already causes G to be non-tensorial. Indeed, it is not possible (without resorting to a particular Gilkey decomposition) to construct an affine connection from G, not even on the embedding space $\Lambda^2 TM \supset A^2 TM$.

A.2.1 Area metric curvature

In A.2we have shown that the bundle of tangent areas $A^2 TM$ over a manifold M is the one of instant relevance to area metric geometry, and that $A^2 T_p M$ is not a linear space, but merely a variety, a polynomial subspace of the vector space $\Lambda^2 T_P M$ of antisymmetric two-tensors. While it is of course possible to equip non-vector bundles with a connection, which is determined in terms of a covariant derivative ∇ on the vector bundle $\Lambda^2 TM$. For n-dimensional area metric manifold (M, Gg), the metric Gg gives rise to the torsion-free Levi-Civita connection, which lifts to the $\Lambda^2 TM$-bundle in the standard way.

To commence, we consider any local coordinate system about some point on the manifold, and let M be n-dimensional complete Reimannian manifold with Riemannian metric g_{ij}, the Levi-Civita connection is given by Chistoffel symbols

$$\Gamma^k_{ij} = \frac{1}{2} g^{kl} \left\{ \frac{\partial g_{il}}{\partial x^i} + \frac{\partial g_{il}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^l} \right\}. \tag{A.6}$$

where g^{ij} is the inverse of g_{ij}. The Riemannian curvature tensors read

$$R^k_{ijl} = \frac{\partial \Gamma^k_{ji}}{\partial x^i} - \frac{\partial \Gamma^k_{il}}{\partial x^j} + \Gamma^k_{ip}\Gamma^p_{jl} - \Gamma^k_{jp}\Gamma^p_{il}. \tag{A.7}$$

a covariant version leads to the definition of the area metric curvature as follows

$$R_{ijkl} = g_{kp}R^p_{ijl} \tag{A.8}$$

The area Ricci tensor is the contraction

$$R_{ik} = g^{jl}R_{ijkl} \tag{A.9}$$

where the scalar curvature is

$$R = g^{ij}R_{ij} \tag{A.10}$$

The given metric g satisfies the hyperbolic geometric flow [1]

$$\frac{\partial^2 g_{ij}}{\partial t^2} = -2R_{ij} \tag{A.11}$$

which is a nonlinear system of second order partial differential equations on the metric.

The hyperbolic geometric flow is an evolution equation on the metric $g_{ij}(t, x)$. The evolution for the metric implies a nonlinear wave equation for the area curvature tensor R_{ijkl}, the Ricci curvature tensor R_{ij} and the scalar curvature K. Under the hyperbolic geometric flow (1), the curvature tensors satisfy the evolution equations

$$\begin{aligned} \frac{\partial^2 R_{ijkl}}{\partial t^2} &= LR_{ijkl} + (lower\ order\ terms) \\ \frac{\partial^2 R_{ij}}{\partial t^2} &= LR_{ij} + (lower\ order\ terms) \\ \frac{\partial^2 R}{\partial t^2} &= LR + (lower\ order\ terms) \end{aligned} \tag{A.12}$$

Moreover, it's worth mentioning here that the area curvature tensor satisfies:

1) Skew symmetry $R_{ijkl} = -R_{jikl} = -R_{ijlk}$

2) Interchange symmetry $R_{ijkl} = -R_{klij}$

3) Cyclicity $R_{ijkl} + R_{iklj} + R_{iljk} = 0$

The last property is often written as $R_{i[jkl]} = 0$ where the bracket denotes the antisymmetric part on the indicated indices.

A.3 Area metric under Hyperbolic geometric flow

In this chapter, we will restrict our study to the geometry of manifolds equipped with an arbitrary algebraic curvature map, making essential use of the Gilkey decomposition. More precisely, we will commence by considering the induced-metric area metrics Gg. In this section we will study the area metric under the hyperbolic geometric flow, which is a very natural tool to understand the wave character of the metrics and wave phenomenon of the curvatures. We start with an arbitrary induced-metric area metric Gg given as follows

$$Gg_{ijkl} = g_{ik}g_{jl} - g_{il}g_{jk}. \tag{A.13}$$

differentiating with respect to t gives

$$\frac{\partial Gg_{ijkl}}{\partial t} = g_{ik}\frac{\partial g_{jl}}{\partial t} + \frac{\partial g_{ik}}{\partial t}g_{jl} - \frac{\partial g_{il}}{\partial t}g_{jk} - g_{il}\frac{\partial g_{jk}}{\partial t}. \tag{A.14}$$

once again we differentiate with respect to t, yields

$$\begin{aligned}\frac{\partial^2 Gg_{ijkl}}{\partial t^2} = {} & g_{ik}\frac{\partial^2 g_{jl}}{\partial t^2} + \frac{\partial^2 g_{ik}}{\partial t^2}g_{jl} - \frac{\partial^2 g_{il}}{\partial t^2}g_{jk} - g_{il}\frac{\partial^2 g_{jk}}{\partial t^2} \\ & + (lower\ order\ terms).\end{aligned} \tag{A.15}$$

using A.11, then substituting by A.9 and taking into consideration that g^{ij} is the

inverse of g_{ij}, give the rise to the following

$$\frac{\partial^2 G g_{ijkl}}{\partial t^2} = -2R_{jilk} - 2R_{ijkl} + 2R_{jikl} + 2R_{ijlk} \tag{A.16}$$

The skew symmetry and interchange symmetry properties of the tensor R_{ijkl} previously given in **??** leads to the following hyperbolic geometric flow

$$\frac{\partial^2 G g_{ijkl}}{\partial t^2} = -8R_{ijkl} \tag{A.17}$$

From the relation A.9 together with A.10 one can figure out that

$$\begin{aligned} R_{ijkl} &= R g_{ik} g_{jl} \\ -R_{ijkl} &= R g_{il} g_{jk} \end{aligned} \tag{A.18}$$

which leads to

$$\begin{aligned} \frac{\partial^2 G g_{ijkl}}{\partial t^2} &= -4R[g_{ik} g_{jl} - g_{il} g_{jk}] \\ &= -4RG g_{ijkl} \end{aligned} \tag{A.19}$$

and this give the notation

$$R_{ijkl} = \frac{R}{2} G g_{ijkl} \tag{A.20}$$

APPENDIX A

The COIL dateset

The following tables contain the number of the features points of each graph in the COIL-20 database. The first two tables (A.1, A.2) stands for the number of nodes of the graphs $(G_i, i = 1, ..., 10)$ of the first ten objects, where Table A.1 gives the first 36 poses $(Pi, i = 1, ...36)$ of each graph and Table A.2 gives the next 36 poses $(Pi, i = 37, ...72)$. While the next two tables (A.3, A.4) show the number of nodes of the graphs $(G_i, i = 11, ..., 20)$of the second ten objects with the same arrangement for Table A.3 and Table A.4 as it was for Table A.1 and Table A.2 respectively.

With the same arrangements, Tables(A.5, A.6, A.7 and A.8) show the number of edges of each graph; as well as Tables(A.9, A.10, A.11 and A.12) give the number of triangular faces of each graph.

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
P1	54	59	48	78	60	53	76	53	99	67
P2	48	56	44	72	60	51	73	48	98	72
P3	52	53	43	71	60	49	75	49	96	61
P4	42	51	45	65	60	47	69	47	94	75
P5	47	55	41	64	60	45	57	45	88	68
P6	57	54	40	59	64	45	62	41	81	64
P7	60	58	41	55	69	43	69	38	87	66
P8	66	61	38	54	66	44	57	40	88	60
P9	58	64	39	51	67	38	60	41	94	58
P10	65	63	41	49	68	37	63	42	94	54
P11	60	64	40	52	70	42	65	39	96	60
P12	65	69	48	50	75	39	54	36	100	52
P13	54	65	51	58	82	40	59	37	105	49
P14	55	60	56	52	84	54	62	30	109	50
P15	52	52	67	50	95	67	54	27	109	40
P16	53	54	77	51	96	77	58	26	100	40
P17	56	49	92	50	90	77	63	27	100	37
P18	56	52	107	45	91	91	64	34	94	39
P19	62	42	94	42	82	94	67	29	83	39
P20	60	46	84	44	76	84	63	29	98	42
P21	63	43	70	55	86	68	67	32	97	41
P22	62	47	70	50	94	60	72	32	113	40
P23	54	44	55	48	84	61	68	33	119	49
P24	52	45	53	51	76	48	66	33	103	51
P25	53	40	47	52	74	42	68	40	97	57
P26	42	42	42	46	77	39	69	44	95	57
P27	44	44	39	48	70	38	80	42	90	68
P28	58	58	40	47	71	39	74	48	96	63
P29	51	51	39	44	70	41	73	46	96	63
P30	52	52	39	51	70	40	78	48	97	66
P31	64	54	36	45	74	38	82	48	98	71
P32	51	52	39	48	68	43	73	50	99	74
P33	58	65	43	46	76	46	74	54	99	71
P34	54	62	44	55	72	50	81	55	103	75
P35	55	68	39	27	76	51	72	57	109	68
P36	54	61	43	34	70	51	79	49	107	75

Table A.1: Number of the feature points of the first 10 objects (Poses 1 to 36) of the COIL-20 database

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
P37	57	64	45	49	73	47	84	58	106	70
P38	50	60	41	48	70	50	78	49	96	64
P39	58	62	43	48	68	45	82	48	103	63
P40	85	74	41	50	74	53	84	52	88	61
P41	57	66	37	49	76	51	82	46	87	61
P42	62	74	36	42	80	50	83	47	90	62
P43	61	71	43	40	75	47	83	48	96	65
P44	71	70	38	42	73	41	74	43	102	65
P45	72	75	48	44	74	40	80	39	93	64
P46	73	90	47	38	78	51	70	38	95	58
P47	81	88	49	45	71	49	63	37	91	55
P48	71	90	50	44	77	48	68	37	105	56
P49	68	84	53	49	85	53	67	30	101	58
P50	68	87	60	51	94	62	73	31	102	50
P51	57	78	64	52	103	68	78	29	92	48
P52	58	68	77	47	95	75	78	31	95	41
P53	62	68	78	44	84	89	79	28	98	42
P54	64	78	77	42	85	105	73	32	93	42
P55	65	64	78	47	87	99	74	28	89	38
P56	65	63	84	49	89	87	84	30	95	40
P57	63	59	64	47	92	84	85	40	103	41
P58	57	67	58	55	84	74	84	35	112	41
P59	65	66	50	56	80	62	83	39	108	49
P60	69	67	48	58	70	60	82	34	106	49
P61	68	63	48	56	75	49	83	43	99	48
P62	72	63	45	60	71	47	88	39	96	59
P63	77	58	39	61	69	45	83	45	91	59
P64	64	65	40	60	65	42	86	46	87	67
P65	73	56	44	69	71	46	85	48	79	61
P66	62	58	44	65	66	50	81	51	85	64
P67	58	55	38	73	69	50	83	45	83	63
P68	54	57	43	75	68	50	83	53	87	72
P69	52	58	37	75	62	46	83	49	91	76
P70	48	52	36	74	65	48	75	48	92	70
P71	45	50	44	78	62	47	81	53	99	67
P72	49	57	45	78	61	49	81	52	105	71

Table A.2: Number of the feature points of the first 10 objects (Poses 37 to 72) of the COIL-20 database

	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}	G_{19}	G_{20}
P1	78	99	53	110	66	40	143	31	42	79
P2	83	97	53	114	63	45	127	32	43	82
P3	90	91	57	112	65	39	126	33	46	82
P4	91	86	54	100	76	50	135	34	40	71
P5	89	98	63	95	70	45	123	34	44	82
P6	92	100	62	92	71	43	123	34	41	77
P7	92	93	59	87	68	46	131	40	40	81
P8	90	101	58	86	66	48	129	38	47	80
P9	93	100	52	86	75	42	137	42	47	67
P10	94	109	58	85	63	41	136	50	53	78
P11	91	102	53	76	65	46	143	44	54	80
P12	90	105	57	80	74	44	134	43	54	75
P13	79	94	61	87	66	45	146	49	62	68
P14	89	98	51	94	66	41	125	45	69	70
P15	88	99	56	101	70	36	130	47	74	75
P16	84	99	48	114	65	41	142	45	80	65
P17	81	99	44	107	69	47	142	43	99	80
P18	71	104	42	103	67	41	140	44	106	72
P19	74	100	43	110	69	43	138	49	100	74
P20	70	113	52	113	68	46	146	44	99	70
P21	74	110	47	110	70	43	133	49	88	68
P22	84	101	48	111	69	46	148	49	76	69
P23	75	107	59	102	69	39	151	52	71	71
P24	75	103	49	97	73	44	142	58	65	68
P25	86	107	50	88	70	44	137	46	60	71
P26	94	104	57	87	68	40	137	47	55	68
P27	93	104	61	92	70	41	133	44	50	66
P28	89	106	65	100	71	43	139	47	46	69
P29	95	114	64	104	63	43	152	38	44	68
P30	95	105	68	101	70	38	145	34	39	72
P31	97	101	59	99	59	43	155	36	38	66
P32	105	98	61	110	71	41	147	36	38	75
P33	99	106	48	111	62	42	143	33	42	73
P34	94	102	50	119	69	40	140	32	38	67
P35	94	102	48	126	72	40	135	26	41	72
P36	100	104	47	123	68	48	150	29	39	71

Table A.3: Number of the feature points of the second 10 objects (Poses 1 to 36) of the COIL-20 database

	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}	G_{19}	G_{20}
P37	104	101	45	117	68	49	148	32	37	69
P38	96	106	53	121	73	45	139	29	39	72
P39	98	100	50	116	67	45	140	31	46	69
P40	90	106	51	109	69	43	130	29	49	73
P41	98	104	57	99	69	44	136	28	43	74
P42	95	110	58	85	73	43	132	30	48	73
P43	95	101	58	93	70	38	143	33	42	80
P44	99	108	60	86	65	42	132	33	49	77
P45	92	95	71	84	63	40	136	34	47	70
P46	91	105	76	91	73	38	146	34	54	75
P47	92	101	69	85	66	41	137	32	52	77
P48	83	100	72	81	64	38	136	32	53	76
P49	79	106	62	97	60	30	135	29	66	78
P50	72	11	54	109	64	40	138	36	80	83
P51	83	105	56	94	68	39	166	32	80	84
P52	69	105	56	108	72	40	135	35	90	83
P53	71	105	61	96	76	36	142	37	91	83
P54	68	111	62	100	69	44	142	42	97	88
P55	72	105	64	90	75	37	142	35	97	91
P56	79	105	63	112	75	39	133	31	91	86
P57	76	107	58	97	71	37	141	29	75	82
P58	80	114	56	93	71	34	145	38	68	80
P59	82	115	63	86	72	41	146	33	57	70
P60	75	115	69	84	67	37	147	38	57	73
P61	76	115	76	79	76	43	143	33	57	73
P62	82	116	70	84	72	41	149	36	48	70
P63	87	111	69	83	73	40	150	29	45	72
P64	85	112	77	82	70	34	143	33	46	75
P65	97	111	67	83	69	41	151	36	46	79
P66	88	116	59	84	73	35	148	30	41	78
P67	95	108	63	89	64	37	145	35	43	76
P68	94	111	59	96	68	43	140	26	42	80
P69	99	105	60	105	62	39	145	33	41	78
P70	106	93	59	111	72	40	139	31	44	74
P71	93	98	57	115	66	41	136	30	44	76
P72	87	107	55	116	73	45	134	28	42	71

Table A.4: Number of the feature points of the second 10 objects (Poses 37 to 72) of the COIL-20 database

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
P1	138	163	129	215	168	144	217	140	285	188
P2	124	153	118	200	167	134	206	131	284	203
P3	137	143	116	198	168	130	213	134	276	168
P4	109	136	121	181	167	125	195	124	271	206
P5	122	147	106	176	167	120	157	121	253	190
P6	150	148	105	163	181	120	173	108	233	176
P7	160	158	109	148	194	114	193	99	252	180
P8	176	162	99	146	187	116	157	105	254	162
P9	153	173	102	137	190	101	167	107	270	155
P10	172	169	109	130	194	98	174	110	272	145
P11	159	175	108	136	199	113	181	98	276	160
P12	169	189	132	131	213	105	149	89	285	136
P13	138	176	140	154	235	106	162	92	299	131
P14	140	165	154	138	239	149	172	74	313	131
P15	134	140	185	133	271	188	146	61	313	106
P16	132	147	215	138	274	215	159	62	286	104
P17	144	133	256	135	259	217	172	66	281	96
P18	142	141	300	120	259	256	178	85	261	101
P19	161	112	268	109	236	265	185	69	234	98
P20	154	122	238	115	218	234	174	69	278	111
P21	163	114	196	147	244	188	186	79	270	107
P22	162	123	193	132	267	164	201	80	319	105
P23	140	114	149	129	240	167	187	85	341	127
P24	135	118	143	139	216	129	183	82	297	135
P25	136	104	127	141	211	110	186	105	280	152
P26	145	108	112	124	220	103	191	115	275	151
P27	140	109	102	128	200	99	225	110	260	182
P28	140	155	106	124	201	102	209	124	279	169
P29	152	134	101	115	200	108	207	122	277	171
P30	170	138	103	136	200	105	218	127	278	179
P31	171	149	94	117	210	99	233	130	283	196
P32	133	141	103	125	193	114	207	135	288	205
P33	155	181	115	121	218	122	209	149	284	197
P34	142	172	118	148	204	135	230	149	297	209
P35	143	188	103	66	216	137	206	158	316	188
P36	141	167	114	85	201	139	225	130	308	209

Table A.5: Number of Edges of the first 10 objects (Poses 1 to 36) of the COIL-20 database

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
P37	148	178	120	131	208	126	238	158	306	195
P38	130	164	109	126	200	134	220	134	278	176
P39	153	170	115	127	192	119	231	131	296	175
P40	150	205	110	133	209	142	238	141	254	170
P41	149	179	98	131	215	137	229	124	251	169
P42	163	202	95	111	227	135	235	127	260	169
P43	159	195	115	104	213	127	234	132	275	177
P44	186	190	99	110	205	108	206	115	294	175
P45	187	204	126	117	208	105	222	103	267	172
P46	191	247	124	102	219	137	194	98	273	159
P47	216	245	131	115	200	130	172	95	263	148
P48	184	250	134	116	219	129	187	94	300	149
P49	174	232	141	132	240	144	210	75	284	156
P50	177	238	162	136	269	168	202	77	290	134
P51	150	218	173	138	293	185	218	71	260	127
P52	151	187	211	123	267	205	216	77	265	108
P53	164	187	217	113	236	245	221	69	275	109
P54	169	216	217	108	238	297	204	81	259	107
P55	174	178	219	124	246	278	208	69	256	98
P56	170	174	232	129	250	241	239	74	271	103
P57	165	159	174	124	263	232	242	101	296	108
P58	147	182	161	147	239	201	237	87	322	106
P59	169	178	134	145	228	169	235	99	311	131
P60	184	177	128	154	198	164	229	81	305	131
P61	178	173	128	151	212	133	236	109	282	127
P62	189	170	120	162	200	123	249	100	275	158
P63	199	155	101	162	193	120	235	119	258	160
P64	165	176	104	163	182	110	245	122	251	182
P65	189	151	116	188	198	122	241	128	227	164
P66	159	157	116	176	182	134	229	134	245	173
P67	149	150	101	200	192	132	234	118	240	174
P68	141	154	114	206	190	135	235	141	252	200
P69	136	158	98	206	172	122	236	134	263	215
P70	126	144	95	202	183	128	209	131	265	195
P71	117	136	117	214	174	126	232	140	288	189
P72	129	157	121	214	173	132	227	136	304	198

Table A.6: Number of Edges of the first 10 objects (Poses 37 to 72) of the COIL-20 database

	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}	G_{19}	G_{20}
P1	214	280	146	308	175	103	402	77	112	213
P2	225	269	143	321	164	120	353	81	114	223
P3	248	258	156	314	171	100	354	81	125	224
P4	250	240	144	281	203	131	377	84	105	194
P5	243	277	168	264	181	117	345	85	118	220
P6	254	281	167	254	186	114	345	85	109	211
P7	254	259	158	241	174	120	364	100	107	222
P8	250	285	155	235	174	124	358	98	128	218
P9	258	278	143	238	198	109	385	107	128	207
P10	261	307	157	235	163	107	380	129	147	214
P11	249	286	143	210	147	120	402	114	146	216
P12	247	293	152	218	192	113	375	112	145	204
P13	218	260	161	244	174	115	409	129	169	183
P14	247	275	135	263	170	105	349	117	191	187
P15	247	279	150	282	184	92	364	122	206	204
P16	232	278	123	320	170	105	398	118	224	175
P17	222	279	114	300	180	118	401	114	279	217
P18	195	294	108	289	175	103	392	115	298	195
P19	206	281	110	306	182	109	387	130	284	202
P20	191	319	139	319	183	121	411	117	277	191
P21	207	313	122	312	185	111	371	131	244	185
P22	232	284	126	312	179	119	417	128	209	184
P23	206	301	155	285	179	101	427	140	195	191
P24	206	288	125	269	190	113	399	156	180	182
P25	237	300	129	243	183	112	385	121	165	191
P26	261	289	150	240	179	103	383	123	150	182
P27	256	289	160	253	185	107	373	114	133	177
P28	245	296	172	279	186	112	393	121	123	184
P29	261	320	172	291	163	111	427	98	118	180
P30	263	292	184	280	181	95	411	85	104	191
P31	266	284	158	273	152	111	437	91	97	177
P32	290	273	165	307	181	106	409	92	99	202
P33	275	298	127	312	161	107	399	85	111	196
P34	261	285	134	337	177	103	394	79	99	178
P35	260	284	127	358	187	103	376	66	106	193
P36	276	291	125	345	176	128	426	73	101	192

Table A.7: Number of Edges of the second 10 objects (Poses 1 to 36) of the COIL-20 database

	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}	G_{19}	G_{20}
P37	288	283	120	329	177	132	413	80	96	184
P38	267	298	144	341	194	117	390	72	101	194
P39	273	281	134	328	173	120	394	78	123	184
P40	252	298	135	306	177	113	365	73	127	197
P41	272	291	153	279	180	116	382	69	112	199
P42	264	310	156	240	189	113	370	73	127	200
P43	264	282	157	257	183	96	400	82	111	219
P44	274	304	162	238	169	110	369	81	126	211
P45	255	266	198	228	164	103	381	87	124	186
P46	250	295	208	251	189	98	409	86	141	199
P47	254	280	188	236	168	105	382	82	136	212
P48	226	284	193	224	168	97	381	79	140	201
P49	217	298	163	272	155	73	379	71	178	214
P50	198	314	143	306	166	102	385	91	219	227
P51	232	298	148	263	176	97	468	79	220	228
P52	192	296	148	303	189	101	374	88	250	228
P53	199	298	161	269	200	89	402	91	254	226
P54	188	313	166	276	182	112	395	105	271	240
P55	200	298	167	246	199	94	401	85	273	251
P56	221	298	162	313	201	98	374	76	255	235
P57	209	301	154	269	190	95	397	70	209	226
P58	219	324	148	261	186	86	409	96	184	215
P59	228	329	166	237	189	104	412	85	151	185
P60	209	327	188	226	176	92	413	99	150	199
P61	210	327	206	215	201	109	399	81	155	195
P62	227	333	190	232	190	103	418	93	128	186
P63	240	315	190	232	195	103	419	73	120	191
P64	230	319	214	227	182	86	403	82	124	204
P65	269	314	185	233	181	102	427	89	124	216
P66	240	329	162	231	193	89	415	72	109	212
P67	262	306	174	244	168	94	405	86	114	204
P68	255	315	161	267	180	112	394	61	111	213
P69	273	298	165	293	163	99	408	79	109	213
P70	296	263	162	313	189	104	392	77	119	199
P71	254	276	156	323	173	106	382	74	119	203
P72	237	299	151	331	194	117	378	68	112	193

Table A.8: Number of Edges of the second 10 objects (Poses 37 to 72) of the COIL-20 database

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
P1	85	105	82	138	109	92	142	88	187	122
P2	77	98	75	129	108	84	134	84	187	132
P3	86	91	74	128	109	82	139	86	181	108
P4	68	86	77	117	108	79	127	78	178	132
P5	76	93	66	113	108	76	101	77	166	123
P6	94	95	66	105	118	76	112	68	153	113
P7	101	101	69	94	126	72	125	62	166	115
P8	111	102	62	93	122	73	101	66	167	103
P9	96	110	64	87	124	64	108	67	177	98
P10	108	107	69	82	127	62	112	69	179	92
P11	100	112	69	85	130	72	117	60	181	101
P12	105	121	85	82	139	67	96	54	186	85
P13	85	112	90	97	154	67	104	56	195	83
P14	86	106	99	87	156	96	111	45	205	82
P15	83	89	119	84	177	122	93	35	205	67
P16	80	94	139	88	179	139	102	37	187	65
P17	89	85	165	86	170	141	110	40	182	60
P18	87	90	194	76	169	166	115	52	168	63
P19	100	71	175	68	155	172	119	41	152	60
P20	95	77	155	72	143	151	112	41	181	70
P21	101	72	127	93	159	121	120	48	174	67
P22	101	77	124	83	174	105	130	49	207	66
P23	87	71	95	82	157	107	120	53	223	79
P24	84	74	91	89	141	82	118	50	195	85
P25	84	65	81	90	138	69	119	66	184	96
P26	90	67	71	79	144	65	123	72	181	95
P27	86	66	64	81	131	62	146	69	171	115
P28	86	98	67	78	131	64	136	77	184	107
P29	96	84	63	72	131	68	135	77	182	109
P30	106	87	65	86	131	66	141	80	182	114
P31	108	96	59	73	137	62	152	83	186	126
P32	83	90	65	78	126	72	135	86	190	132
P33	98	117	73	76	143	77	136	96	186	127
P34	89	111	75	94	133	86	150	95	195	135
P35	89	121	65	40	141	87	135	102	208	121
P36	88	107	72	52	132	89	147	82	202	135

Table A.9: Number of triangulated faces of the first 10 objects (Poses 1 to 36) of the COIL-20 database

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
P37	92	115	76	83	136	80	155	101	201	126
P38	81	105	69	79	131	85	143	86	183	113
P39	96	109	73	80	125	75	150	84	194	113
P40	93	132	70	84	136	90	155	90	167	110
P41	93	114	62	83	140	87	148	79	165	109
P42	102	129	60	70	148	86	153	81	171	108
P43	99	125	73	65	139	81	152	85	180	113
P44	116	121	62	69	133	68	133	73	193	111
P45	116	130	79	74	135	66	143	65	175	109
P46	119	158	78	65	142	87	125	61	179	102
P47	136	158	83	71	130	82	110	59	173	94
P48	114	161	85	73	143	82	120	58	196	94
P49	107	149	89	84	157	92	135	46	184	99
P50	110	152	103	86	176	107	130	47	189	85
P51	94	141	110	87	191	118	141	43	169	80
P52	94	120	135	77	173	131	139	47	171	68
P53	103	120	140	70	153	157	143	42	178	68
P54	106	139	141	67	154	193	132	50	167	66
P55	110	115	142	78	160	180	135	42	168	61
P56	106	112	149	81	162	155	156	45	177	64
P57	103	101	111	78	172	149	158	62	194	68
P58	91	116	104	93	156	128	154	53	211	66
P59	105	113	85	90	149	108	153	61	204	83
P60	116	111	81	97	129	105	148	48	200	83
P61	111	111	81	96	138	85	154	67	184	80
P62	118	108	76	103	130	77	162	62	180	100
P63	123	98	63	102	125	76	153	75	168	102
P64	102	112	65	104	118	69	160	77	165	116
P65	117	96	73	120	128	77	157	81	149	104
P66	98	100	73	112	117	85	149	84	161	110
P67	92	96	64	128	124	83	152	74	158	112
P68	88	98	72	132	123	86	153	89	166	129
P69	85	101	62	132	111	77	154	86	173	140
P70	79	93	60	129	119	81	135	84	174	126
P71	73	87	74	137	113	80	152	88	190	123
P72	81	101	77	137	113	84	147	85	200	128

Table A.10: Number of triangulated faces of the first 10 objects (Poses 37 to 72) of the COIL-20 database

	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}	G_{19}	G_{20}
P1	137	182	94	199	110	64	260	47	71	135
P2	143	173	91	208	102	76	227	50	72	142
P3	159	168	100	203	107	62	229	49	80	143
P4	160	155	91	182	128	82	243	51	66	124
P5	155	180	106	170	112	73	223	52	75	139
P6	163	182	106	163	116	72	223	52	69	135
P7	163	167	100	155	107	75	234	61	68	142
P8	161	185	98	150	109	77	230	61	82	139
P9	166	179	92	153	124	68	249	66	82	132
P10	168	199	100	151	101	67	245	80	95	137
P11	159	185	91	135	92	75	260	71	93	137
P12	158	189	96	139	119	70	242	70	92	130
P13	140	167	101	158	109	71	264	81	108	116
P14	159	178	85	170	105	65	225	73	123	118
P15	160	181	95	182	115	57	235	76	133	130
P16	149	180	76	207	106	65	257	74	145	111
P17	142	181	71	194	112	72	259	72	181	138
P18	125	191	67	187	109	63	253	72	193	124
P19	133	182	68	197	114	67	250	82	185	129
P20	122	207	88	207	116	76	266	74	179	122
P21	134	204	76	203	116	69	239	83	157	118
P22	149	184	79	202	111	74	270	80	134	116
P23	132	195	97	184	111	63	277	89	125	121
P24	132	186	77	173	118	70	258	99	116	115
P25	152	194	80	156	114	69	249	76	106	121
P26	168	186	94	154	112	64	247	77	96	115
P27	164	186	100	162	116	67	241	71	84	112
P28	157	191	108	180	116	70	255	75	78	116
P29	167	207	109	188	101	69	276	61	75	113
P30	169	188	117	180	112	58	267	52	66	120
P31	170	184	100	175	94	69	283	56	60	112
P32	186	176	105	198	111	66	263	57	62	128
P33	177	193	80	202	100	66	257	53	70	124
P34	168	184	85	219	109	64	255	48	62	112
P35	167	183	80	233	116	64	242	41	66	122
P36	177	188	79	223	109	81	277	45	63	122

Table A.11: Number of triangulated faces of the second 10 objects (Poses 1 to 36) of the COIL-20 database

	G_{11}	G_{12}	G_{13}	G_{14}	G_{15}	G_{16}	G_{17}	G_{18}	G_{19}	G_{20}
P37	185	183	76	213	110	84	266	49	60	116
P38	172	193	92	221	122	73	252	44	63	123
P39	176	182	85	213	107	76	255	48	78	116
P40	163	193	85	198	109	71	236	45	80	125
P41	175	188	97	181	112	73	247	42	70	126
P42	170	201	99	156	117	71	239	44	80	128
P43	170	182	100	165	114	59	258	50	70	140
P44	176	197	103	153	105	69	238	49	79	135
P45	164	172	127	145	102	64	246	54	78	117
P46	160	191	133	161	117	61	264	53	88	125
P47	163	180	120	152	103	65	246	51	85	136
P48	144	185	122	144	105	60	246	48	88	126
P49	139	193	102	176	96	44	245	43	113	137
P50	127	204	90	198	103	63	248	56	140	145
P51	150	194	93	170	109	59	303	48	141	145
P52	124	192	93	196	118	62	240	54	161	146
P53	129	194	101	174	125	54	261	55	164	144
P54	121	203	105	177	114	69	254	64	175	153
P55	129	194	104	157	125	58	260	51	177	161
P56	143	194	100	202	127	60	242	46	165	150
P57	134	195	97	173	120	59	257	42	135	145
P58	140	211	93	169	116	53	265	59	117	136
P59	147	215	104	152	118	64	267	53	95	116
P60	135	213	120	143	110	56	267	62	94	127
P61	135	213	131	137	126	67	257	49	99	123
P62	146	218	121	149	119	63	270	58	81	117
P63	154	205	122	150	123	64	270	45	76	120
P64	146	208	138	146	113	53	261	50	79	130
P65	173	204	119	151	113	62	277	54	79	138
P66	153	214	104	148	121	55	268	43	69	135
P67	168	199	112	156	105	58	261	52	72	129
P68	162	205	103	172	113	70	255	36	70	134
P69	175	194	106	189	102	61	264	47	69	136
P70	191	171	104	203	118	65	254	47	76	126
P71	162	179	100	209 1	08	66	247	45	76	128
P72	151	193	97	216	122	73	245	41	71	123

Table A.12: Number of triangulated faces of the second 10 objects (Poses 37 to 72) of the COIL-20 database

References

Arendt, W. & Warma, M. (2003). Dirichlet and neumann boundary conditions: What is in between? *Journal of Evolution Equations*, *3*, 119 – 135.

Atkins, J. E., Boman, E. G., & Hendrickson, B. (1998). A spectral algorithm for seriation and the consecutive ones problem. *SIAM J. Comput.*, *28(1)*, 297–310.

Bakry, D., Coulhon, T., Ledoux, M., & Saloff-Coste, L. (1995). Sobolev inequalities in disguise. *Indiana Univ. Math. J.*, *44(4)*, 1033 – 1074.

Barlow, M. T. (1998). Diffusions on fractals. *Lecture Notes Math (1690), Springer*, 1–121.

Barreno, M. (2004). Spectral methods for image clustering. *Tech-Report CS 218B, U.C. Berkeley*.

Belkin, M. & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, *14*.

Belkin, M. & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comp.*, *15(6)*, 1373 – 1396.

Bertalmio, M., Cheng, L. T., Osher, S., & Sapiro, G. (2001). Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, *174*, 759–780.

Biggs, N. L. (1993). *Algebraic graph theory*. Cambrridge University Press.

Bobkov, S. & Ledoux, M. (1997). Poincar´e's inequalities and talagrand's concentration phenomenon for the exponential distribution. *Probab. Theory Related Fields*, *107(3)*, 383 – 400.

Bougleux, S. & Elmoataz, A. (2005). Image smoothing and segmentation by graph regularization. *LNCS 3656*, 745–752.

Boykov, Y. & Huttenlocher, D. (1999). A new bayesian framework for object recognition. *Proceeding of IEEE Computer Society Conference on CVPR*, *2*, 517–523.

Brand, M. (2004). From subspaces to submanifolds. *(Technical Report 2004 -134). Mitsubishi Electric Research Laboratories*.

Burges, C. J. C. (2005). Geometric methods for feature extraction and dimensional reduction. *Data mining and knowledge discovery handbook: A complete guide for practitioners and researchers. Kluwer Academic Publishers.*

Carcassoni, M. & Hancock, E. (2001). Weighted graph-matching using modal clusters. *In Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, 260 – 269.

Carcassoni, M. & Hancock, E. R. (2003). Correspondence matching with modal clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 1609 – 1615.

Chan, T., Osher, S., & Shen, J. (Feb. 2001). The digital tv filter and nonlinear denoising. *IEEE Trans. Image Process*, *10(2)*, 231–241.

Chan, T. & Shen, J. (2000). Variational restoration of non-flat image features: Models and algorithms. *SIAM J. Appl. Math.*, *61*, 1338–1361.

Cheng, L., Burchard, P., Merriman, B., & Osher, S. (September 2000). Motion of curves constrained on surfaces using a level set approach. Technical report, UCLA CAM Technical Report (00-32).

Chung, F. & Yau, S. (1995). Eigenvalues of graphs and sobolev inequalities. *Combinatorics, Probability and Computing*, *4*, 11 – 25.

Chung, F. R. (1997). Spectral graph theory. *in Proc. CBMS Regional Conf. Ser. Math.*, *92*, 1–212.

Chung, F. R. K., Faber, V., & Manteuffel, T. A. (1994). An upper bound on the diameter of a graph from eigenvalues associated with its laplacian. *SIAM J. Discrete Math.*, *7(3)*, 443 – 457.

Chung, F. R. K., Grigor'yan, A., & Yau, S.-T. (1996). Upper bounds for eigenvalues of the discrete and continuous laplace operators. *Adv. Math.*, *117(2)*, 165 – 178.

Chung, F. R. K., Grigor'yan, A., & Yau, S.-T. (1997). Eigenvalues and diameters for manifolds and graphs. *In Tsing Hua lectures on geometry & analysis (Hsinchu, 1990–1991), Internat. Press, Cambridge, MA*, 79 – 105.

Clarke, C. J. S. (1970). On the global isometric embedding of pseudo-riemannian manifolds. *Proceedings of Royal Society of London. A . [314]*, 417 – 428.

Collatz, L. & Sinogowitz, U. (1957). Spektren endlicher grafen. *Abh. Math. Sem. Univ. Hamburg*, *21*, 63 – 77.

Costa, J. & Hero, A. O. (2004). Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. on Signal Process.*, *52*, 2210 – 2221.

Coulhon, T. (1992). Sobolev inequalities on graphs and on manifolds. *In Harmonic analysis and discrete potential theory (Frascati, 1991), Plenum, New York*, 207 – 214.

Coulhon, T. (1996a). Espaces de lipschitz et in´egalit´es de poincar´e. *J. Funct. Anal.*, *136(1)*, 81 – 113.

Coulhon, T. (1996b). Ultracontractivity and nash type inequalities. *J. Funct. Anal.*, *141(2)*, 510 – 539.

Coulhon, T. & Grigor'yan, A. (1997). On-diagonal lower bounds for heat kernels and markov chains. *Duke Math. J.*, *89(1)*, 133 – 199.

Cox, T. & Cox, M. (1994). *Multidimensional Scaling*. Chapman-Hall.

Cristianini, N., Taylor, J., & Kandola, J. (2001). Spectral kernel methods for

clustering. *In NIPS*, 649 – 655.

de Verdi‘ere, Y. C. (1998). Spectres de graphes. *Societe Mathematique De France*.

Dhillon, I. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *In KDD ’01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press.*, 269 – 274.

Diaconis, P. & Saloff-Coste, L. (1996). Nash inequalities for finite markov chains. *J. Theoret. Probab.*, *9(2)*, 459 – 510.

Donoho, D. L. & Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, *100(10)*, 5591–5596.

Doob, M., Sachs, H., & Cvetkovic’, D. (1995). Spectra of graphs - theory and applications. *Johann Ambrosius Barth Verlag, Heidelberg-Leipzig, the third revised and enlarged edition, Books.*

Dubuisson, M. & Jain, A. (1994). A modified hausdorff distance for object matching. (pp. 566–568).

Duda, R. O. & Hart, P. E. (1973). Pattern classification and scene analysis. *Wiley*.

Early, E. (1999). On the euler characteristic. *MIT Undergraduate Journal of Mathematics*, *1*, 39 – 48.

Elmoataz, A., Lezoray, O., & Bougleux, S. (2008). Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Trans. on Image Processing*, *17(7)*, 1047 – 1060.

Emms, D., Wilson, R., & Hancock, E. (2007). Graph embedding using quantum commute times. *Graph-Based Representations in Pattern Recognition*, *LNCS 4538*, 371 – 382.

Fiedler, M. (1993). A geometric approach to the laplacian matrix of a graph. *Combinatorial and Graph-Theoretical Problems in Linear Algebra*, 73 – 98.

Folland, G. B. (1995). *Introduction to partial differential equations*. Princeton University Press.

Friedman, J. (1993). Some geometric aspects of graphs and their eigenfunctions. *Duke Mathematical Journal*, *69(3)*, 487 – 525.

Friedman, J. & Tillich, J.-P. (12 Aug 2004a). Calculus on graphs. *arXiv:cs.DM/0408028*.

Friedman, J. & Tillich, J.-P. (2004b). Wave equations for graphs and the edge based laplacian. *Pacific Journal of Mathematics, 216(2)*, 229 – 266.

Gauss, C. (1900). *Allgemeine Flächentheorie(Translated from Latin)*. W. Engelmann.

Gilbarg, D. & Trudinger., N. S. (1983). *Elliptic Partial Differential Equations of Second Order*. Springer-Verlag.

Gilkey, P. B. (1984). *Invariance theory, heat equation, and the index theorem*. Mathematics Lecture Series.

Grigor'yan, A. (2001). Heat kernels on manifolds, graphs and fractals. *European Congress of Mathematics*, *I*, 393–406.

Grigor'yan, A. (2006). Heat kernels on weighted manifolds and applications. *Cont. Math.*, *398*, 93 – 191.

Grone, R. (1991). On the geometry and laplacian of a graph. *Linear Algebra and Appl.*, *1501*, 167 – 178.

Ham, J., Lee, D. D., Mika, S., & Scholkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. *Proceedings of the Twenty-First International Conference on Machine Learning (ICML). Banff, Alberta, Canada.*, 369 – 376.

Han, L., Wilson, R. C., & Hancock, E. R. (2010). A supergraph-based generative model. *ICPR*, 1566–1569.

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H. (2005). Face recognition using laplacianfaces. *IEEE. Trans. Pattern Anal. Mach. Intell.*, *27*, 328–340.

Hein, M., Audibert, J., & Von Luxburg, U. (2005). From graphs to manifolds-weak and strong pointwise consistency of graph laplacians. (pp. 470 – 485).

Heinonen, J., Kilpelainen, T., & Martio, O. (1993). Nonlinear potential theory of degenerate elliptic equations. *Oxford University Press, Oxford*.

Heut, B. & Hancock, E. R. (2002). Relational object recognition from large structural libraries. *Pattern Recognition*, *32*, 1895–1915.

Horaud, R. & Sossa, H. (1995). Polyhedral object recognition by indexing. *Pattern Recognition*, *28*, 1855–1870.

Hotelling, H. (1933). Analysis of complex statistical variables in principal components. *J. Educational Psychology*, *24*, 417 – 441.

Hurt, N. E. (2000). *Mathematical physics of quantum wires and devices*. Kluwer Academic Publishers, Dordrecht.

Huttenlocher, D., Klanderman, G., & Rucklidge, W. (1993). Comparing images using the hausdorff distance. *IEEE. Trans. Pattern Anal. Mach. Intell.*, *15*, 850–863.

Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: a review. *ACM Computing Surveys*, *31(3)*, 264 – 323.

Jolliffe, I. T. (1986). Principal component analysis. *New York: Springer-Verlag.*

Kannan, R., Vempala, S., & Vetta, A. (2000). On clusterings: Good, bad, and spectral. *In Proceedings of the 41st Annual Symposium on the Foundation of Computer Science , IEEE Computer Society*, 367 – 380.

Kimmel, R., Malladi, R., & Sochen, N. (2000). Images as embedding maps and minimal surfaces: Movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, *39(2)*, 111–129.

Kluger, Y., Basri, R., Chang, J., & Gerstein, M. (2003). Spectral biclustering of microarray data:coclustering genes and conditions. *Genome Res.*, *13(4)*, 703 – 716.

Kosinov, S. & Caelli, T. (2002a). Inexact multisubgraph matching using graph eigenspace and clustering models. *9th International Workshop on Structural and Syntactic Pattern Recognition*, *LNCS 2396*, 133 – 142.

Kosinov, S. & Caelli, T. (2002b). Inexact multisubgraph matching using graph eigenspace and clustering models. *In Proc. Joint IAPR Int. Workshops SSPR and SPR*, 133 – 142.

Kruskal, J. B. & Wish, M. (1978). Multidimensional scaling. *Sage Publications. Beverly Hills. CA.*

Kulis, B., Basu, S., Dhillon, I., & Mooney, R. (2005). Semi-supervised graph clustering: a kernel approach. *In ICML '05: Proceedings of the 22nd international conference on Machine learning, New York, NY, USA, ACM Press.*, 457 – 464.

Lafon, S. (2004). *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University.

Lebanon, G. & Lafferty, J. D. (2004). Hyperplane margin classifiers on the multinomial manifold. *ICML*.

Lim, B. P., Montenegro, J. F., & Santos, N. L. (14 Aug 2008). Eigenvalues estimates for the p-laplace operator on manifolds. *arXiv:0808.2028v1 [math.DG]*.

Lopez-Perez, L., Deriche, R., & Sochen, N. (2004). The beltrami flow over triangulated manifolds. *LNCS 3117*, 135–144.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60(2)*, 91–110.

Luo, B., Cross, A., & Hancock, E. R. (1999). Corner detection via topographic analysis of vector potential. *Pattern Recognition Letters*, *20*, 635 – 650.

Luo, B. & Hancock, E. R. (2001). Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, *23(10)*, 1120–1136.

Luo, B., Wilson, R. C., & Hancock, E. R. (2003). Spectral embedding of graphs. *Pattern Recogintion*, *36*, 2213–2230.

Luxburg, U. V. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17(4)*, 395 – 416.

Luxburg, U. V., Belkin, M., & Bousquet, O. (2004). Consistency of spectral clustering. *Technical Report 134, Max Planck Institute for Biological Cybernetics*.

Luxburg, U. V., Bousquet, O., & Belkin, M. (2005). Limits of spectral clustering. *In Lawrence K. Saul, Yair Weiss, and leon Bottou, editors, Advances in Neural Information Processing Systems (NIPS) 17, MIT Press, Cambridge, MA*.

Meila, M. & Shi, J. (2000). Learning segmentation by random walks. *In NIPS*, 873 – 879.

Memoli, F., Sapiro, G., & Osher, S. (January 2002). Solving variational problems and partial differential equations, mapping into general target manifolds. Technical report, UCLA CAM Technical Report (02-04).

Merris, R. (1994). Laplacian matrices of graphs: a survey. *Linear Algebra Appl.*, *197-198*, 143 – 176.

Merris, R. (1995). A survey of graph laplacians. *Linear Algebra Appl.*, *39*, 19 – 31.

Merris, R. & Grone, R. (1994). The laplacian spectrum of a graph ii. *SIAM J. on Discrete Math.*, *7*, 221 – 229.

Mohar, B. (1991). The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, *2*, 871 – 898.

Mohar, B. (1992). Laplace eigenvalues of graphs - a survey. *Discrete Math.*, *109*, 171 – 183.

Mohar, B. (1997). Some applications of laplace eigenvalues of graphs. *Graph Symmetry: Algebraic Methods and Applications, NATO ASI Series*, *C 497*, 227 – 275.

Nash, J. F. (1954). C1-isometric imbeddings. *Ann. Math., [60]*, 383 – 396.

Nash, J. F. (1956). The imbedding problem for riemannian manifolds. *Ann. Math., [63]*, 20 – 63.

Newman, M., Watts, D., & Strogatz, S. (2002). Random graph models of social networks. *Proc. Natl. Acad. Sci. USA*, *99*, 2566 – 2572.

Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14, Cambridge, MA, MIT Press.*

Osher, S. & Shen, J. (2000). Digitized pde method for data restoration. *In Analytical-Computational methods in Applied Mathematics, E. G. A. Anastassiou, Ed. New York: Chapman & Hall/CRC*, 751–771.

Paccanaro, A., C. C.-C. J. & Saqi, M. (2003). Spectral clustering of protein sequences. *In International Joint Conference on Neural Networks*, *4*, 3083

– 3088.

Pekalaska, E., Duin, R. P. W., & Paclik, P. (2006). Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, *39(2)*, 189 – 208.

Pekalska, E. & Haasdonk, B. (2009). Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE transactions on pattern analysis and machine intelligence, 31(6)*, 1017 – 1032.

Rahimi, A. & Recht, B. (2004). Clustering with normalized cuts is clustering with a hyperplane. *Statistical Learning in Computer Vision*.

Riesen, K., Neuhaus, M., & Bunke, H. (2007). Graph embedding in vector spaces by means of prototype selection. *Graph-Based Representations in Pattern Recognition*, *LNCS 4538*, 383 – 393.

Robles-Kelly, A. & Hancock, E. R. (2002). A graph-spectral approach to correspondence matching. *International Conference on Pattern Recognition.*

Robles-Kelly, A. & Hancock, E. R. (2007). A riemannian approach to graph embedding. *Pattern Recognition*, *40(3)*, 1042 – 1056.

Roweis, S. T. & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science 22 December*, *290 (5500)*, 2323 – 2326.

Sachs, H., Cvetkovic, D., & Doob, M. (1980). *Spectra of graphs*. Academic Press.

Saerens, M., Fouss, F., Yen, L., & Dupont, P. (2004). The principal components analysis of a graph, and its relationships to spectral clustering. *ECML*, 371 – 383.

Saloff-Cost, L. (1997). Lectures on finite markov chains. *In Lectures on probability theory and statistics (Saint-Flour, 1996), Springer, Berlin*, 301 – 413.

Sapiro, G. (January 2001). *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press.

Saul, L. K., Weinberger, K. Q., Sha, F., Ham, J., & Lee, D. D. (2005). Spectral methods for dimensionality reduction. *Semisupervised learning. MIT Press.*

Scholkopf, B., Smola, A., & K.-R. Muller, K.-R. (1998). Nonlinear component

analysis as a kernel eigenvalue problem. *Neural Computation*, *10(5)*, 1299 – 1319.

Scott, G. & Longuett-Higgins, H. (1991). An algorithm for associating the features of two images. *Proceedings of the Royal Society of London Series B-Biological*, *244*, 21 – 26.

Sengupta, K. & Boyer, K. (1998). Modelbase paritioning using property matrix spectra. *Computer Vision and Imaging Understanding*, *70*, 177–196.

Shapiro, L. S. & Brady, J. M. (1992). Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, *10*, 283 – 288.

Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE. PAMI*, *22*, 888–905.

Shokoufandeh, A. & Dickinson, S. (2001). A unified framework for indexing and matching hierarchical shape structures. *Lecture Notes in Computer Science*, *2059*, 67 – 84.

Shokoufandeh, A., Dickinson, S. J., Siddiqi, K., & Zucker, S. W. (1999). Indexing using a spectral encoding of topological structure. *CVPR*, 2491–2497.

Shokoufandeh, S., Macrini, D., Dickinson, S. J., Siddiqi, K., & Zucker, S. W. (2005). Indexing hierarchical structures using graph spectra. *IEEE Trans. Pattern Anal. Mach. Intell.*, *27(7)*, 1125 – 1140.

Smola, A. & Kondor, R. (2003a). Kernels and regularization on graphs. *Conference on Learning Theory, COLT/KW*.

Smola, A. J. & Kondor, I. R. (2003b). Kernels and regularization on graphs. *Proc. Annual Conf. Computational Learning Theory (B. Schölkopf and M. K. Warmuth, eds.)*, *LNCS 2726*, 144 – 158.

Smola, A. J., Schölkopf, B., & Müller, K.-R. (1998). The connection between regularization operators and support vector kernels. *Neural Networks*, *11*, 637 – 649.

Sochen, N., Deriche, R., & Lopez-Perez, L. (2003a). The beltrami flow over implicit manifolds. In *ICCV*.

Sochen, N., Deriche, R., & Lopez-Perez, L. (Barcelone 2003b). Variational beltrami flows over manifolds. In *IEEE ICIP 2003*.

Sochen, N., Deriche, R., & Lopez-Perez, L. (June 2003c). Variational beltrami flows over manifolds. Technical report, INRIA Resarch Report 4897.

Sochen, N. & Kimmel, R. (2001). Stereographic orientation diffusion. *in proceedings of the 4th Int. Conf. on Scale-Space, Vancouver Canada*, *October*.

Sochen, N., Kimmel, R., & Malladi, R. (1996). From high energy physics to low level vision. *Report, LBNL, UC Berkeley, LBNL 39243, August, Presented in ONR workshop, UCLA*, *Sept. 5*.

Sochen, N., Kimmel, R., & Malladi, R. (1998). A general framework for low level vision. *IEEE Trans. on Image Processing*, *7*, 310–318.

Sochen, N. & Zeevi, Y. (1998). Representation of colored images by manifolds embedded in higher dimensional non-euclidean space. *Proc. IEEE ICIP'98*, *Chicago*.

Spillmann, B., M., N., Bunke, H., Pekalaska, E., & Duin, R. P. W. (2006). Transforming strings to vector spaces using prototype selection, structural, syntactic, and statistical pattern recognition. *Joint IAPR International Workshops, SSPR 2006 and SPR 2006*, 287 – 296.

Spivak, M. (1979). *A Comprehensive Introduction to Differential Geometry*. Publish or Parish 2nd ed, vol. 1-5, Houston.

Srivastava, A. (2004). Mixture density mercer kernels: A method to learn kernels directly from data. *In SDM*.

Stillwell, J. (1974). *Mathematics and its History*. Springer-Verlag, New York.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science 290*, *2319*.

Torsello, A. & Hancock, E. R. (2007). Graph embedding using tree edit-union. *Pattern Recognition*, *40*, 1393 – 1405.

Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Patt. Anal. Mach. Intell.*, *10*, 695–703.

Verma, D. & Meila, M. (2005). A comparison of spectral clustering algorithms. *Technical Report, Department of CSE University of Washington Seattle, WA 98195 -2350*.

Weinberger, K. Q. & Saul, L. K. (2006). Unsupervised learning of image man-

ifolds by semidefinite programming. *International Journal of Computer Vision*, *70*, 77 – 90.

Whitney, H. (1936). Differentiable manifolds. *Ann. of Math., 37(2)*, 645 – 680.

Wilson, R. C., Hancock, E. R., & Luo, B. (2005). Pattern vectors from algebraic. *IEEE. Trans. Pattern Anal. Mach. Intell.*, *27*, 1112–1124.

Xiao, B., Hancock, E., & HangYu (2010). Manifold embedding for shape analysis. *Neurocomputing [73]*, 1606 – 1613.

Xiao, B. & Hancock, E. R. (2004). Heat kernel, riemannian manifolds and graph embedding. *LNCS 3138*, 198 –206.

Xiao, B. & Hancock, E. R. (2006). Trace formula analysis of graphs. *SSPR/SPR*, 306–313.

Xiao, B., Hancock, E. R., & Wilson, R. C. (2009). Graph characteristics from the heat kernel trace. *Pattern Recognition, 42(11)*, 2589 – 2606.

Xiao, B., Wilson, R. C., & Hancock, E. R. (2005). Characterising graphs using the heat kernel. *in Proc. BMVC*.

Xu, L. & King, I. (2001). A pca approach for fast retrieval of structural patterns in attributed graphs. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, *31(5)*, 812 – 817.

Xu, R. & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16(3)*, 645 – 678.

Yamasaki, M. (1986). Ideal boundary limit of discrete dirichlet functions. *Hiroshima Math. J.*, *16(2)*, 353 – 360.

Yau, S. T. & Schoen, R. M. (1988). *Differential Geometry*. Science Publication Co. (in Chinese).

Young, G. & Householder, A. S. (1938). Disscussion of a set of points in terms of their mutual distances. *Psychometrika*, *3*, 19–22.

Zha, H., He, X., Ding, C., Gu, M., & Simon, H. (2001). Spectral relaxation for k-means clustering. *In NIPS*, 1057 – 1064.

Zhang, Z. & Zha, H. (2004). Principle manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, *26*, 313 – 338.

Zhou, D. & Schölkopf, B. (2004). A regularization framework for learning from graph data. *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, 132–137.

Zhou, D. & Schölkopf, B. (2005). Regularization on discrete spaces. *LNCS 3663*, 361–368.

Zhou, D. & Schölkopf, B. (2006). Discrete regularization. *Semi-Supervised Learning, O. Chapelle, B. Schölkopf, and A. Zien, Eds. MIT Press, Cambridge, MA.*, 221–232.

Zhu, X., Kandola, J. S., Ghahramani, Z., & Lafferty, J. D. (2004). Nonparametric transforms of graph kernels for semi-supervised learning. *NIPS*.